



TESIS KI142502

Prediksi Wave-induced Liquefaction dengan Artificial Neural Network dan Wide Genetic Algorithm

Dwi Kristianto
NRP. 5115201041

DOSEN PEMBIMBING
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.
Bilqis Amaliah, S.Kom., M.Kom.

PROGRAM MAGISTER
BIDANG KEAHLIAN KOMPUTASI CERDAS DAN VISUALISASI
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017

Prediksi Wave-induced Liquefaction dengan Artificial Neural Network dan Wide Genetic Algorithm

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)
di
Institut Teknologi Sepuluh Nopember

oleh :
Dwi Kristianto
NRP. 5115201041

Tanggal Ujian: 3 Januari 2017
Periode Wisuda: Maret 2017

Disetujui Oleh :

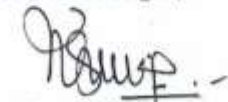
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.
NIP. 19751220 20011220 02


(Pembimbing 1)

Bilqis Amaliah, S.Kom., M.Kom.
NIP. 19750914 20011220 02


(Pembimbing 2)

Dr.Eng. Nanik Suciati, S.Kom., M.Kom.
NIP. 19710428 19941220 01


(Penguji 1)

Anny Yuniarti, S.Kom., M.Comp.Sc
NIP. 19810622 20050120 02


(Penguji 2)

Dini Adni Navastara, S.Kom., M.Sc.
NIP. 19851017 20150420 01


(Penguji 3)

an. Direktur Program Pasca Sarjana,
Asisten Direktur Program Pasca Sarjana



Prof. Dr. Ir. Tri Wicajaja, M.Eng
NIP. 19611021 19860310 01

Direktur Program Pasca Sarjana,

Prof.Ir. Djauhar Manfaat, M.Sc, Ph.D
NIP. 19601202 19870110 01

[Halaman ini sengaja dikosongkan]

*...carilah dahulu
Kerajaan Allah dan kebenaranNya,
maka semuanya itu
akan ditambahkan kepadamu.
...yang mustahil bagimu, mungkin bagiKu.
- Jesus Christ*

Sebuah persembahan untuk:
**Dessri Yani Dewi Purwanti
Delphi Ivana Kristianto
Debian Imanuel Kristianto**

[Halaman ini sengaja dikosongkan]

UCAPAN TERIMA KASIH

Puji Tuhan, akhirnya laporan akhir Tesis S2 ini bisa diselesaikan dengan baik. Tiada kata yang dapat dipanjatkan selain Puji Syukur atas segala berkat dan hikmat yang telah Dia curahkan selama masa studi dan penyusunan Tesis S2 ini. Terima Kasih Ya Bapa di Sorga. Dalam suasana ucapan syukur itu, penulis mengucapkan terima kasih kepada semua pihak yang memberikan bantuan baik moral maupun materiil, secara khusus kepada :

1. Bapak Prof. Dr. Ketut Buda Artana, ST., M.Sc, yang telah memberikan tantangan dan kesempatan untuk melanjutkan studi S2. Terima kasih untuk bimbingan, bantuan dan perhatian yang telah tercurah.
2. Bapak I Made Ariana, ST., MT., Dr.MarSc. yang selalu meluangkan waktu dan perhatiannya untuk mendukung proses studi S2, terutama setiap awal bulan.
3. Bapak A.A. Bagus Dinariyana DP, ST., MES, Ph.D. yang telah membangkitkan semangat untuk studi lanjut terutama pada masa-masa awal perkuliahan S2 serta menjadi sahabat diskusi akademik. Terima kasih untuk ide-ide penelitiannya.
4. Bapak Kriyo Sambodho, ST., M.Eng, Dr.Eng yang selalu setia menemani selama proses perkuliahan S2, penulisan buku tesis dan memberikan dataset yang diperlukan dalam tesis ini.
5. Ibu Dr.Eng. Chastine Fatichah, S.Kom., M.Kom. dan ibu Bilqis Amaliah, S.Kom., M.Kom. yang telah cukup sabar memberikan bimbingan dan tuntunan serta menjadi sahabat diskusi selama proses penyusunan tesis ini.
6. Bapak Dr. Yeyes Mulyadi, ST., MSc., bapak Dr. Dimas Widhi Handani, ST., M.Sc., bapak Dr. I Putu Sindhu Asmara, ST. MT. dan rekan peneliti dari Jurusan Teknik Kelautan dan Laboratorium RAMS, Jurusan Teknik Sistem Perkapalan, FTK ITS. Terima kasih atas asupan semangatnya.
7. Bapak Dwi Suasti Antara, ST. (kandidat MT) dan Charlie Angels (Dilla, Ucik, Emmy dan Putri). Terima kasih atas bantuan menyelesaikan beberapa beban pekerjaan dan administrasi.

8. Subhan Nooriansyah, S.Kom., Didih Rizki Chandranegara, S.Kom. dan Rudianto, ST. yang telah membantu banyak hal dalam penyusunan buku tesis ini. Terima kasih telah hadir membantu pada saat-saat kritis.
9. Seluruh adik-adik di Lab. RAMS Jurusan Teknik Sistem Perkapalan, FTK-ITS, antara lain: Galih, Adi, Afro, Sita, Cakra, Libri, Dante, Ical, dkk yang tidak bisa saya sebutkan satu persatu. Terima kasih telah menunjukkan suasana akademis nan gayeng guyub.
10. Terakhir, terima kasih dari lubuk hati terdalam untuk keluarga kecilku tercinta, istriku Dessri Yani Dewi Purwanti, anakku *mbarep* Delphi Ivana Kristianto dan anakku *ragil* Debian Imanuel Kristianto. Terima kasih telah mengorbankan waktu kebersamaan kita, mencurahkan perhatian dan memelihara cinta kita selama ini.

Satu ucapan terima kasih tidak mencukupi untuk mewakili sedemikian besar rasa syukur dalam hati terdalam. Untuk bapak/ibu/sdr/i yang belum tersebut dalam halaman ini, sekali lagi penulis mengucapkan banyak terima kasih atas bantuan, motivasi serta perhatian yang telah tercurah. Kiranya Tuhan memberkati kita semua. Amin.

Prediksi Wave-induced Liquefaction dengan Artificial Neural Network dan Wide Genetic Algorithm

Nama Mahasiswa : Dwi Kristianto
NRP : 5115201041
Pembimbing : Dr.Eng. Chastine Fatichah, S.Kom, M.Kom
Bilqis Amaliah, S.Kom, M.Kom

ABSTRAK

Fenomena *Liquefaction* telah menjadi perhatian peneliti sejak era 1970an, mulai dari pemodelan analitik, pemodelan dengan Finite Element Method (FEM) sampai beragamnya jenis pengujian di laboratorium. Salah satu bidang dalam penelitian *liquefaction* adalah memprediksi lokasi kedalaman tanah tempat terjadinya *liquefaction*. Kerumitan pemodelan analitik, kebutuhan untuk melakukan validasi laboratorium dan inspeksi lapangan, telah membuka peluang untuk pengembangan pemodelan yang mudah, praktis, murah sekaligus valid.

Dalam penelitian ini, pemodelan Artificial Neural Network (ANN) digunakan untuk memprediksi kedalaman maksimum *liquefaction*. ANN dipilih karena ANN mampu memodelkan interaksi komputasi paralel pada otak melalui proses *learning* terhadap data. Metode Back Propagation (BP) adalah metode pelatihan yang paling banyak digunakan, walaupun memiliki kelemahan mudah terjebak dalam *local optimum* dan tidak stabil. Untuk mengatasi masalah tersebut, metode optimasi Genetic Algorithm (GA) digunakan dalam proses pelatihan ANN. GA adalah metode optimasi yang menirukan proses evolusi, seleksi, rekombinasi dan mutasi yang terjadi di alam.

Walaupun telah banyak digunakan, GA memiliki beberapa kelemahan yaitu *premature convergence* dan *local optimum*. Dalam penelitian ini dilakukan modifikasi terhadap GA, antara lain: operasi seleksi *Wide Tournament*, operasi rekombinasi *BLX- α Multi-Parent*, operasi mutasi *Aggregate Mate Pool* dan operasi *Direct Mutation-Recombination*. Modifikasi GA bertujuan untuk meningkatkan keragaman populasi, memperluas cakupan pencarian solusi dan menemukan *global optimum* lebih mudah dan cepat. Global optimum GA adalah konfigurasi ANN terbaik dengan error prediksi MdAPE terkecil.

Kata kunci: *Wave-induced Liquefaction, Algoritma Genetika Lebar, Jaringan Saraf Tiruan, seleksi Wide Tournament, rekombinasi BLX- α Multi Induk, mutasi Aggregate Mate Pool, Direct Mutation-Recombination*

[Halaman ini sengaja dikosongkan]

Prediction of Wave-induced Liquefaction

Using Artificial Neural Network and Wide Genetic Algorithm

Student Name : Dwi Kristianto
NRP : 5115201041
Supervisor : Dr.Eng. Chastine Fatichah, S.Kom, M.Kom
Bilqis Amaliah, S.Kom, M.Kom

ABSTRACT

Liquefaction phenomenon has been the attention of researchers since the 1970s, ranging from analytical modeling, Finite Element Method (FEM) modeling to various types laboratory testing. One of its research field is to predict the depth where soil liquefaction occurred. The hassle of analytic modeling, repetitive laboratory testing and expensive field inspections, has opened up opportunities to develop simple, practical, inexpensive and valid modeling.

In this research, Artificial Neural Network (ANN) regression modeling is used to predict the maximum depth of liquefaction. ANN selected among other Artificial Intelligence methods because its ability to model the interaction of brain parallel computing through iterative learning process of inputted data. Back Propagation (BP) is one of widely used training method for ANN. Although it has some weaknesses such as easily trapped in local optimum and unstable results, it is most widely used one. To overcome BP weaknesses, Genetic Algorithm (GA) is being used for ANN learning process. GA is optimization method that mimics the process of evolution, selection, recombination and mutation that occurs in nature.

Although GA is a well known method, it also has some weaknesses, such as premature convergence and local optimum. In this research, GA is heavily modified to overcome its weaknesses. The modifications are Wide Tournament selection, Multi-Parent BLX- α recombination, Aggregate mutation and Direct Mutation-Recombination operation. The modifications aim to increase the population diversity, expanding the coverage area of solution search and find global optimum solution more easily and quickly. Global optimum solution of GA is best ANN configuration which has smallest MdAPE prediction errors.

Keywords: *Wave-induced Liquefaction, Artificial Neural Network, Wide Genetic Algorithm, Wide Tournament selection, BLX- α Multi-Parent recombination, Aggregate mutation, Direct Mutation-Recombination*

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

ABSTRAK	vii
ABSTRACT	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
BAB 1 PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah.....	3
1.3. Tujuan dan Manfaat.....	3
1.4. Kontribusi Penelitian	4
1.5. Batasan Masalah	4
1.6. Sistematika Penulisan.....	4
BAB 2 STUDI LITERATUR.....	5
2.1. Wave-induced Liquefaction	5
2.2. Artificial Neural Network (ANN)	7
2.2.1. Metode Pelatihan Back Propagation (BP).....	9
2.2.2. Parameter dan Kinerja ANN.....	10
2.2.3. Kesalahan Prediksi	12
2.2.4. Normalisasi Dataset dan Denormalisasi Hasil Prediksi.....	13
2.2.5. Fungsi Aktivasi.....	13
2.2.6. Normalisasi Model Prediksi	14
2.2.7. Validasi Silang.....	14
2.3. Genetic Algorithm (GA).....	15
2.3.1. Pembangkitan Populasi Acak	17
2.3.2. Operasi Seleksi N-Tournament	18
2.3.3. Operasi Rekombinasi BLX- α	18
2.3.4. Operasi Rekombinasi Multi Induk	19
2.3.5. Operasi Mutasi Bilangan Riil	19
2.3.6. Steady State GA	20
2.3.7. Operasi Penggantian dalam SSGA	21
2.4. Hibridasi ANN-GA	21
BAB 3 METODOLOGI PENELITIAN.....	23
3.1. Studi Literatur.....	23
3.2. Desain Model Sistem.....	24
3.2.1. Pembangkitan Dataset	24
3.2.2. Pelatihan ANN menggunakan GA	25
3.2.3. Prediksi Liquefaction menggunakan ANN-WGA.....	34
3.3. Pembuatan Perangkat Lunak	35
3.4. Pengujian Sistem (Empiris).....	35
3.5. Analisa Hasil Pengujian	36
BAB 4 UJICoba DAN PEMBAHASAN.....	39
4.1. Lingkungan Uji Coba	39
4.2. Spesifikasi Dataset.....	39

4.3.	Skenario Uji Coba Prediksi Liquefaction	41
4.4.	Hasil Uji Coba Skenario 1	43
4.4.1.	Digram Scattered Uji Coba Skenario 1	43
4.4.2.	Diagram Konvergensi Uji Coba Skenario 1	48
4.5.	Hasil Uji Coba Skenario 2	53
4.5.1.	Diagram Scattered Uji Coba Skenario 2	53
4.5.2.	Diagram Konvergensi Uji Coba Skenario 2	58
4.6.	Analisa dan Pembahasan Hasil Uji Coba	63
BAB 5 KESIMPULAN DAN SARAN		67
5.1.	Kesimpulan	67
5.2.	Saran	67
DAFTAR PUSTAKA		69
LAMPIRAN 1 HASIL UJI COBA SKENARIO 1: BOBOT ANN		73
LAMPIRAN 2 HASIL UJI COBA SKENARIO 2: BOBOT ANN		77

DAFTAR GAMBAR

Gambar 1.1 Soil Liquefaction	1
Gambar 1.2 Back Propagation terjebak dalam Local Optimum.....	2
 Gambar 2.1 Grafik F_s terhadap kedalaman (z).....	6
Gambar 2.2 Ilustrasi sistem syaraf di otak (Shen, 2013).....	8
Gambar 2.3 Proses Pelatihan ANN	9
Gambar 2.4 Ilustrasi Global dan Local Optimum (Olson, 2013).....	10
Gambar 2.5 Normalisasi model prediksi.....	14
Gambar 2.6 Ilustrasi partisi data latih dan uji dengan metode K-fold.....	14
Gambar 2.7 Bentuk gen GA hasil proses <i>encoding</i>	16
Gambar 2.8 Iterasi metode optimasi Genetic Algorithm.....	16
Gambar 2.9 Jarak Euclidean.....	17
Gambar 2.10 Eksploitasi dan eksplorasi BLX- α	19
Gambar 2.11 Pemodelan Steady State GA.....	20
 Gambar 3.1 Tahapan Penelitian	23
Gambar 3.2 Alur Penelitian.....	24
Gambar 3.3 Alur pelatihan ANN menggunakan metode optimasi GA.....	26
Gambar 3.4 Konsep Operasi Seleksi Wide Tournament.....	27
Gambar 3.5 Alur Seleksi Wide Tournament	28
Gambar 3.6 Operasi Rekombinasi BLX- α Multi-Induk.....	30
Gambar 3.7 Operasi mutasi beberapa allele dan dalam blok	32
Gambar 3.8 Alur operasi rekombinasi dan mutasi langsung.....	32
Gambar 3.9 Konfigurasi ANN untuk prediksi Liquefaction, $5_{(i)}-9_{(h)}-1_{(o)}$ di sebelah kiri dan $5_{(i)}-9_{(h)}-7_{(h)}-1_{(o)}$ di sebelah kanan.	36
 Gambar 4.1 Diagram scattered pada skenario 1 dataset 1	44
Gambar 4.2 Diagram scattered pada skenario 1 dataset 2.....	44
Gambar 4.3 Diagram scattered pada skenario 1 dataset 3.....	45
Gambar 4.4 Diagram scattered pada skenario 1 dataset 4.....	45
Gambar 4.5 Diagram scattered pada skenario 1 dataset 5.....	46
Gambar 4.6 Diagram scattered pada skenario 1 dataset 6.....	47
Gambar 4.7 Diagram scattered pada skenario 1 dataset 7.....	47
Gambar 4.8 Diagram konvergensi pada skenario 1 dataset 1.....	48
Gambar 4.9 Diagram konvergensi pada skenario 1 dataset 2.....	49
Gambar 4.10 Diagram konvergensi pada skenario 1 dataset 3.....	50
Gambar 4.11 Diagram konvergensi pada skenario 1 dataset 4.....	50
Gambar 4.12 Diagram konvergensi pada skenario 1 dataset 5.....	51
Gambar 4.13 Diagram konvergensi pada skenario 1 dataset 6.....	51
Gambar 4.14 Diagram konvergensi pada skenario 1 dataset 7.....	52
Gambar 4.15 Diagram scattered pada skenario 2 dataset 1	54
Gambar 4.16 Diagram scattered pada skenario 2 dataset 2.....	54

Gambar 4.17 Diagram scattered pada skenario 2 dataset 3	55
Gambar 4.18 Diagram scattered pada skenario 2 dataset 4	56
Gambar 4.19 Diagram scattered pada skenario 2 dataset 5	56
Gambar 4.20 Diagram scattered pada skenario 2 dataset 6	57
Gambar 4.21 Diagram scattered pada skenario 2 dataset 7	57
Gambar 4.22 Diagram konvergensi pada skenario 2 dataset 1	58
Gambar 4.23 Diagram konvergensi pada skenario 2 dataset 2	59
Gambar 4.24 Diagram konvergensi pada skenario 2 dataset 3	60
Gambar 4.25 Diagram konvergensi pada skenario 2 dataset 4	60
Gambar 4.26 Diagram konvergensi pada skenario 2 dataset 5	61
Gambar 4.27 Diagram konvergensi pada skenario 2 dataset 6	61
Gambar 4.28 Diagram konvergensi pada skenario 2 dataset 7	62
Gambar 4.29 Diagram konvergensi pada skenario 1 dan 2	64

DAFTAR TABEL

Tabel 2.1 Konstanta Relative Density D_r	7
Tabel 2.2 Konstanta Earth Pressure at rest (K_0)	7
Tabel 2.3 Formula praktis parameter ANN.....	12
Tabel 2.4 Konstanta parameter GA.....	17
 Tabel 3.1 Contoh sebagian dataset Liquefaction.....	 25
Tabel 3.2 Spesifikasi lingkungan pengembangan	35
Tabel 3.3 Dataset untuk Uji Coba	35
 Tabel 4.2 Spesifikasi Lingkungan Pengujian.....	 39
Tabel 4.3 Spesifikasi Dataset	39
Tabel 4.4 Paramater dan Konstanta Umum.....	41
Tabel 4.5 Parameter dan Konstanta BP.....	41
Tabel 4.6 Parameter dan Konstanta GA	42
Tabel 4.7 Parameter dan Konstanta WGA	42
Tabel 4.8 Ringkasan Hasil Pengujian Skenario 1	43
Tabel 4.9 Ringkasan Hasil Pengujian Skenario 2	53
Tabel 4.10 Perbandingan Hasil Pengujian Skenario 1 dan 2.....	63

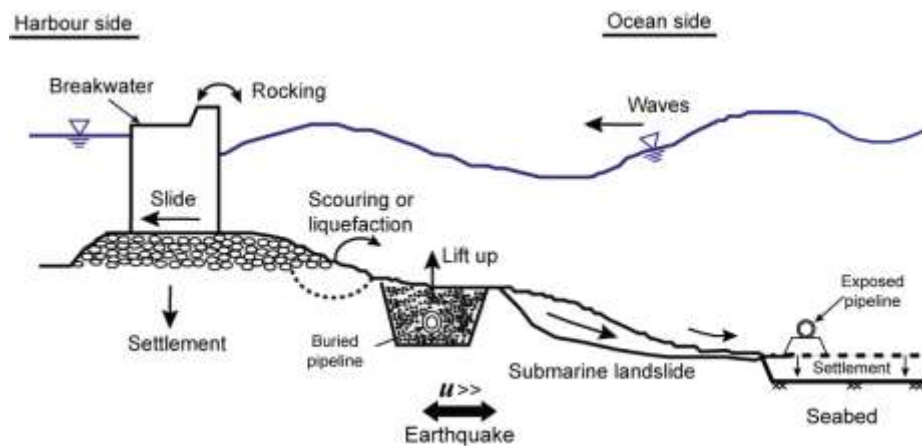
[Halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Penelitian tentang *Liquefaction* telah mulai dilakukan sejak era tahun 1970-an (Sumer & Fredsoe, 2005), mulai dari pemodelan analitik, analisa menggunakan Finite Element Method (FEM) bahkan melakukan pengujian laboratorium (Sekiguchi, et al., 2004) (Sekiguchi, et al., 2001). *Wave-induced Liquefaction* adalah kejadian bertambahnya beban tekanan di atas tanah yang melebihi kapasitas tekanan partikel tanah akibat beban siklik gelombang di perairan sekitar pantai. Bila Liquefaction terjadi, maka struktur di pantai dapat mengalami kegagalan pondasi dan mengakibatkan kerugian yang besar (Gambar 1.1).

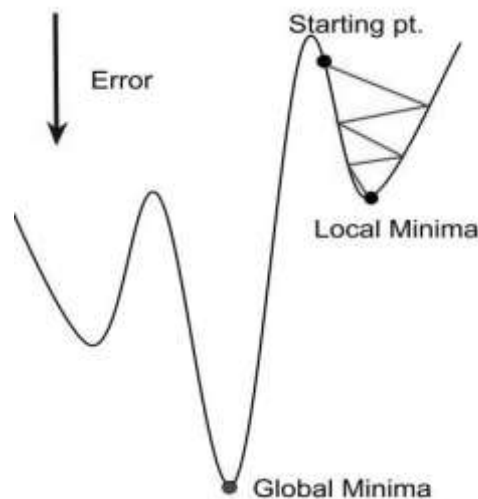


Gambar 1.1 Soil Liquefaction

Salah satu bidang penelitian Liquefaction adalah melakukan pemodelan untuk memprediksi Liquefaction, yaitu lokasi kedalaman terjadinya Liquefaction dan kapan terjadinya. Pemodelan secara analitik yang dilakukan cukup rumit dan membutuhkan validasi pengujian di laboratorium dan juga membutuhkan inspeksi di lapangan yang mahal. Hal ini memberikan ruang untuk pengembangan pemodelan yang mudah, praktis, murah sekaligus valid. Cha et al. mengusulkan pemodelan untuk memprediksi Liquefaction dengan Artificial Neural Network (ANN) dan Simple Genetic Algorithm (SGA) (Cha, et al., 2009).

Implementasi Artificial Intelligence (AI) utamanya ANN dalam bidang prediksi Liquefaction tergolong masih baru sehingga memberikan ruang cukup luas untuk pengembangan lebih lanjut. ANN dipilih dari sekian banyak metode klasifikasi karena ANN mampu memodelkan interaksi komputasi paralel pada otak melalui proses belajar (*learning*)

terhadap data (Gorunescu, 2010). Salah satu metode *learning* data yang banyak digunakan dalam ANN adalah Back Propagation (BP). Pencarian konfigurasi ANN menggunakan BP dapat terjebak pada kondisi *local optimum* (Gambar 1.2). Iterasi awal BP dimulai dari pembangkitan bilangan acak untuk setiap bobot koneksi di ANN. Hal ini tidak akan menjadi masalah bila titik awal iterasi dekat dengan *global optimum*, yang biasanya sangat jarang terjadi (Han & Kamber, 2006). Untuk itu diperlukan metode pencarian solusi yang lebih luas jangkauan pencariannya.



Gambar 1.2 Back Propagation terjebak dalam Local Optimum

Metode optimasi Genetic Algorithm (GA) adalah metode optimasi yang menirukan proses evolusi, seleksi, rekombinasi dan mutasi yang terjadi di alam. Pertama kali diusulkan oleh John Henry Holland di tahun 1975, GA telah digunakan di berbagai disiplin ilmu pengetahuan dan teknologi (Holland, 2012). GA juga digunakan untuk melakukan proses pelatihan (*learning*) bagi ANN. Cha et.al menggunakan Simple Genetic Algorithm dalam pelatihan ANN untuk memprediksi liquefaction (Cha, et al., 2009). GA yang digunakan dalam penelitian ini telah dimodifikasi untuk meningkatkan akurasi prediksi ANN, antara lain: operasi seleksi *Wide Tournament*, operasi rekombinasi (*crossover*) *Modified BLX- α Multi-Parent* dan operasi mutasi *Aggregate Mate Pool*.

Modifikasi GA yang dilakukan bertujuan untuk menghindari kondisi *local optimum* dan *premature convergence* dengan cara memperluas cakupan pencarian solusi (eksplorasi) dan mendapatkan solusi *global optimum*. Operasi Wide Tournament adalah operasi seleksi yang memilih secara otomatis gen terburuk dalam populasi. Gen terburuk tersebut dapat mengurangi *pressure* (tekanan) dalam operasi seleksi sehingga dapat berakibat pada meningkatnya *diversity*

(keragaman) populasi. Operasi rekombinasi Modified BLX- α Multi-Parent bertujuan untuk memperluas rentang nilai dan meningkatkan kombinasi nilai allele setiap offspring. Kedua hal tersebut dapat meningkatkan daya eksplorasi pada ruang solusi. Operasi mutasi Aggregated Value Mate Pool menggunakan nilai-nilai allele *aggregate* setiap gen yang terpilih dalam *Mate Pool* sehingga operasi mutasi dapat lebih adaptif terhadap tingkat konvergensi setiap generasi. Operasi pemeriksaan anti duplikasi memastikan bahwa setiap individu dalam populasi adalah individu yang unik, sehingga tingkat keragaman dalam populasi selalu terjaga dengan baik dan dapat pula meningkatkan daya eksplorasi pada ruang solusi.

Jenis populasi GA yang digunakan adalah Steady State GA (SSGA) dengan operasi *Worst Replacement (WR)* (Lozano, et al., 2008). Operasi WR mengakibatkan perubahan gen terburuk dalam setiap generasi karena digantikan oleh offspring yang lebih baik akibat operasi rekombinasi dan mutasi. Perubahan gen terburuk ini sangat penting karena akan memberikan dampak pada operasi seleksi Wide Tournament di generasi selanjutnya, yaitu perubahan arah pencarian solusi *global optimum*. SSGA membatasi jumlah gen dalam populasi selalu konstan dan berukuran kecil, antara 20-40 individu, sehingga proses *sorting* (pengurutan), operasi anti duplikasi dan operasi WR dapat dilakukan sangat cepat.

1.2. Perumusan Masalah

Rumusan masalah dalam penelitian ini dapat dijabarkan sebagai berikut, antara lain:

1. Bagaimana melakukan pemodelan prediksi Wave-induced Liquefaction dengan metode Artificial Neural Network (ANN)
2. Bagaimana melakukan optimasi bobot koneksi di ANN menggunakan metode Wide Genetic Algorithm (WGA) untuk mendapatkan kesalahan prediksi sekecil mungkin
3. Bagaimana menghitung kinerja sistem dalam memprediksi Wave-induced Liquefaction

1.3. Tujuan dan Manfaat

Tujuan penelitian ini adalah melakukan pemodelan regresi dengan metode ANN dan melakukan pelatihan ANN dengan WGA untuk mendapatkan konfigurasi terbaik ANN dengan kesalahan prediksi sekecil mungkin.

1.4. Kontribusi Penelitian

Kontribusi yang diusulkan dalam penelitian ini adalah memodifikasi beberapa operator dalam metode WGA, yaitu operator seleksi *Wide Tournament*, operasi rekombinasi *BLX- α Multi-Parent*, operasi mutasi *Aggregate Mate Pool* dan operasi *Direct Mutation-Recombination*.

1.5. Batasan Masalah

Penelitian ini berfokus pada fenomena Soil Liquefaction yang disebabkan oleh beban siklik gelombang pada perairan di sekitar pantai.

1.6. Sistematika Penulisan

BAB 1 PENDAHULUAN

Bab ini berisi pendahuluan yang menjelaskan latar belakang, perumusan masalah, tujuan, manfaat, kontribusi, batasan masalah dan sistematika penulisan.

BAB 2 STUDI LITERATUR

Bab ini menjelaskan dasar teori dan studi literatur yang berhubungan dengan penelitian yang akan dilakukan. Bab ini berisi Wave-induced Liquefaction, Artificial Neural Network, Genetic Algorithm dan Pengujian Awal ANN-GA.

BAB 3 METODOLOGI PENELITIAN

Bab ini berisi tahapan penelitian yang meliputi dataset, desain metode, skenario ujicoba dan evaluasi yang digunakan dalam penelitian ini.

BAB 4 UJICOBA DAN PEMBAHASAN

Bab ini berisi Bab ini berisi skenario dan lingkungan uji coba serta evaluasi yang dilakukan.

BAB 5 KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang dapat diambil dari penelitian ini beserta saran untuk penelitian selanjutnya.

BAB 2

STUDI LITERATUR

Pada Bab 2 ini akan dijelaskan mengenai Wave-induced Liquefaction, Artificial Neural Network, Genetic Algorithm dan Hibridasi ANN-GA

2.1. Wave-induced Liquefaction

Soil Liquefaction adalah suatu fenomena alam yang biasa mengikuti kejadian gempa bumi (*earthquake-induced liquefaction*) dan gelombang laut (*wave-induced liquefaction*), seperti terlihat pada Gambar 1.1. Pada kondisi ini, tanah jenuh air (*saturated*) akan kehilangan kekuatan gesernya (*shear strength* = τ'). Hilangnya kekuatan geser ini akibat berkurangnya tegangan efektif (σ') hingga mendekati 0, akibat naiknya tekanan air pori tanah (*pore water pressure* = p) pada saat gempa bumi atau akibat tekanan gelombang laut.

$$\tau' = c + \sigma' \tan \varphi \quad (2.1)$$

$$\sigma' = \sigma - p \quad (2.2)$$

Tuna dan Altun memformulasikan kondisi liquefaction dengan formulasi (Tuna & Altun, 2015) :

$$F_s = \frac{CRR}{CSR} \quad (2.3)$$

CRR (Cyclic Resistance Ratio) adalah kekuatan atau daya dukung tanah terhadap beban, sering juga disebut sebagai *liquefaction resistance*. CSR (Cyclic Stress Ratio) adalah beban gelombang terhadap tanah yang dapat mengakibatkan liquefaction. Ishihara menggunakan rumus empirik untuk memformulasikan CRR dan CSR sebagai (Ishihara & Yamazaki, 1984) :

$$CRR = \frac{\tau_{vh,l}}{\sigma'_v} = 0.0019 D_r \frac{1+2 K_0}{3} \quad (2.4)$$

$$CSR = \frac{\tau_{vh}}{\sigma'_v} = \frac{2\pi}{\rho'g} \frac{P_0}{L} e^{-2\pi \frac{z}{L}} \quad (2.5)$$

Liquefaction terjadi bila $F_s \leq 1$, sehingga didapatkan z sebagai:

$$z = - \ln\left(\frac{0.0019 D_r \frac{1+2 K_0}{3}}{\frac{2\pi}{\rho'g} \frac{P_0}{L}}\right) \frac{L}{2\pi} \quad (2.6)$$

$$P_0 = \rho_w g \frac{H}{2} \frac{1}{\cosh(2\pi \frac{h}{L})} \quad (2.7)$$

Keterangan:

D_r = konstanta *relative density*

K_0 = konstanta *earth pressure at rest*

ρ_w = massa jenis air laut (kg/m^3)

ρ' = massa jenis tanah *saturated* (kg/m^3)

g = kecepatan gravitasi (9.81 m/dt^2)

P_0 = pore water pressure (N/m^2)

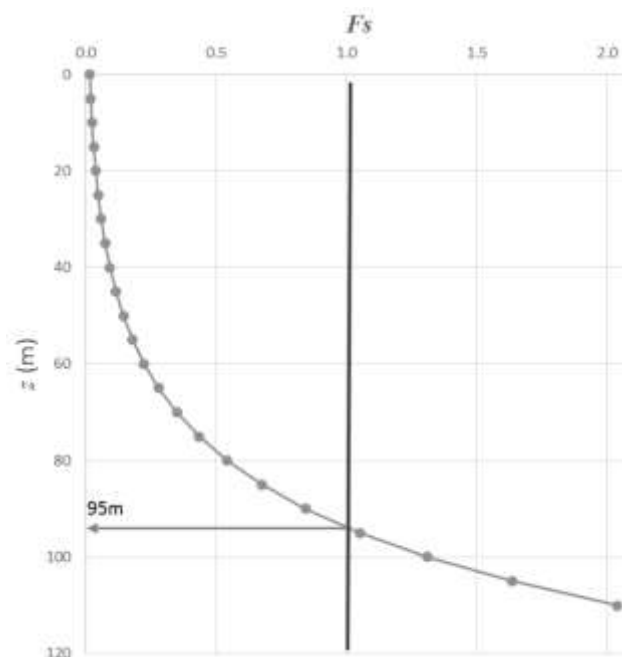
L = panjang gelombang (m)

H = tinggi gelombang (m)

h = kedalaman air (m)

z = kedalaman tanah (m)

Gambar 2.1 adalah grafik F_s terhadap kedalaman tanah (z), berdasarkan data dari Ishihara (Ishihara & Yamazaki, 1984). Dalam gambar tersebut, didapatkan $z = 95\text{m}$ untuk $F_s = 1$, yang berarti liquefaction bisa terjadi sampai kedalaman 95m dibawah *mudline*, untuk kedalaman $z > 95\text{m}$ tidak terjadi liquefaction karena daya dukung tanah masih cukup besar bila dibandingkan dengan beban siklik akibat gelombang laut.



Gambar 2.1 Grafik F_s terhadap kedalaman (z)

Sumer menyarankan konstanta Relative Density (D_r) berdasarkan kategori tanah pada Tabel 2.1 (Sumer, 2014). Jaky menyarankan konstanta Earth Pressure at rest (K_0) berdasarkan jenis tanah pada Tabel 2.2 (Jaky, 1944).

Tabel 2.1 Konstanta Relative Density D_r

Kategori Tanah	D_r
Very Loose	0.00 – 0.15
Loose	0.15 – 0.35
Medium	0.35 – 0.65
Dense	0.65 – 0.85
Very Dense	0.85 – 1.00

Tabel 2.2 Konstanta Earth Pressure at rest (K_0)

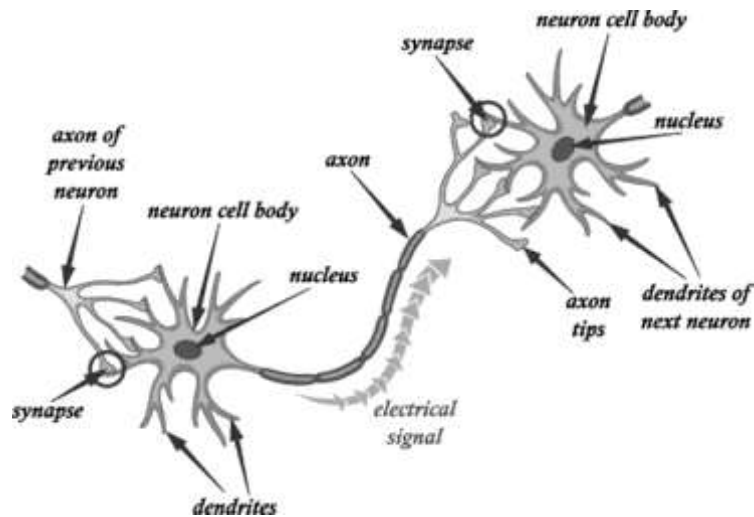
Jenis Tanah	K_0
Sandy Clay	0.25 – 0.42
Silt	0.42 – 0.54
Sand Dense	0.25 – 0.67
Sand Coarse	0.18
Sand Fine-grained	0.33
Rock	0.11 – 0.67

Salah satu dampak Soil Liquefaction (*Liquefaction*) adalah terangkatnya pipa (*pipe floatation*) yang konstruksinya terpendam di dasar laut (*seabed*). Apabila Liquefaction ini berlangsung dalam kurun waktu lama, maka sangat mungkin mengakibatkan kerusakan pipa. Fenomena Liquefaction seringkali diikuti dengan ancaman geoteknik lain, yaitu: rekahan tanah (*lateral spreading*), kelongsoran tanah di dasar laut (*submarine landslide*), hingga penurunan tanah (*settlement*).

2.2. Artificial Neural Network (ANN)

Metode ANN diinspirasi oleh proses pengolahan informasi yang terjadi di otak. Larose mendefinisikan ANN sebagai sebuah sistem pembelajaran (*learning system*) yang kompleks (Larose, 2005). ANN dapat dikatakan sebagai upaya meniru pembelajaran nonlinier (*nonlinear learning*) yang terjadi di jaringan neuron. Jose et al. (Ortiz, et al., 2011) mendefinisikan ANN sebagai pemodelan sangat sederhana dari proses-proses yang terjadi di otak.

Di otak manusia, neuron menggunakan dendrit untuk mendapatkan input dari neuron yang lain dan memberikan respon nonlinier ketika sebuah melebihi suatu batas (*threshold*) tertentu. Setiap neuron tidak berdiri sendiri, melainkan terkoneksi dengan neuron yang lain dan membentuk jaringan neuron yang sangat besar ukurannya. Untuk lebih jelasnya dapat diamati pada Gambar 2.2.



Gambar 2.2 Ilustrasi sistem syaraf di otak (Shen, 2013)

Menurut Ortiz et al., jaringan ANN dapat dikelompokkan berdasarkan beberapa aspek, yaitu (Ortiz, et al., 2011) :

A. Aspek Arsitektur

Arsitektur ANN dapat dibedakan dari ada-tidaknya koneksi feedback dalam ANN. Berikut pembagiannya :

- Arsitektur *Feedforward*. Dalam arsitektur ini tidak ada koneksi balik dari neuron output ke neuron input. ANN Feedforward tidak menyimpan hasil iterasi sebelumnya. Jaringan perceptron adalah salah satu contohnya.
- Arsitektur *Feedback*. Berbeda dengan arsitektur sebelumnya, arsitektur Feedback memiliki koneksi balik dari neuron output menuju neuron input. Arsitektur feedback menyimpan hasil iterasi sebelumnya dan menggunakannya dalam proses latihan. Jaringan Hopfield adalah salah satu contohnya.

B. Aspek Pelatihan

Kemampuan “*belajar*” sebuah ANN diperoleh melalui penerapan metode pelatihan (pembelajaran) tertentu, nampak seperti Gambar 2.3. Berikut ini beberapa jenis metode pelatihan untuk ANN (Ortiz, et al., 2011) :

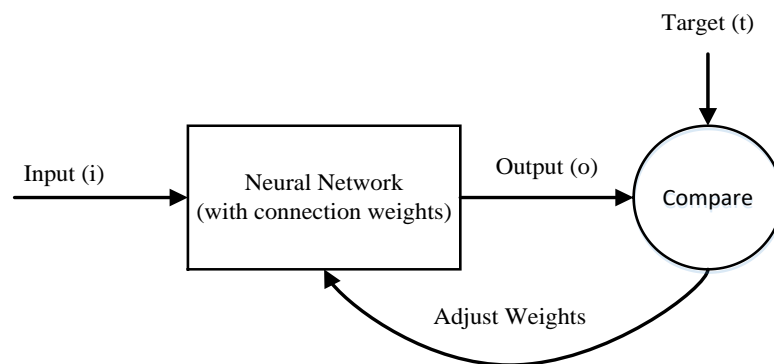
- *Supervised*.
Proses pelatihan ANN dilakukan untuk setiap vektor input sampai ANN mampu “belajar” mengaitkan semua vektor input terhadap semua vektor output yang terkait.

- *Unsupervised.*

Pelatihan ANN yang dilakukan hanya memberikan input vektor saja dan berfokus pada fitur internal keseluruhan input vektor yang telah disediakan.

- *Reinforcement Learning (Reward-penalty Learning).*

Secara umum adalah gabungan dari kedua metode pelatihan sebelumnya, yaitu dengan memberikan reward (penghargaan) bila vektor output yang dihasilkan memberikan hasil yang lebih baik dibandingkan dengan iterasi sebelumnya. Sebaliknya, bila didapatkan hasil yang lebih buruk, punishment diberikan pada ANN berupa pengurangan bobot koneksi.



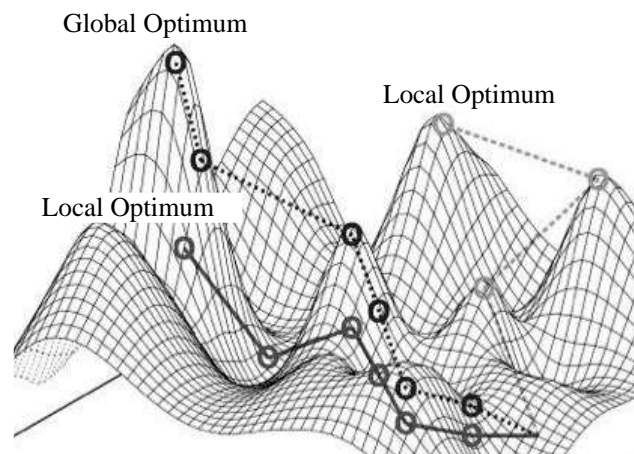
Gambar 2.3 Proses Pelatihan ANN

2.2.1. Metode Pelatihan Back Propagation (BP)

Sampai saat ini, BP adalah salah satu metode learning yang paling banyak dipakai. Tujuan utama BP adalah merubah secara iteratif nilai bobot setiap koneksi dalam ANN untuk mendapatkan error prediksi sekecil mungkin (Park, 2011). Berikut ini beberapa hal yang dapat mempengaruhi kinerja BP, antara lain :

- Kinerja ANN sangat dipengaruhi oleh data yang digunakan dalam proses pelatihan. Bila proses pelatihan menggunakan data yang tidak representatif yaitu tidak mewakili seluruh kombinasi data yang memungkinkan, maka kinerja ANN tidak akan maksimal. Dalam kondisi tersebut, kinerja ANN akan baik hanya pada data latih tersebut dan berkinerja buruk pada keseluruhan dataset.
- Kadangkala data yang digunakan dalam proses pelatihan adalah data yang mirip sama, sehingga ANN gagal mengenali jarak antara data. Dalam kondisi ini, masing-masing data tersebut saling mempengaruhi (interferensi) output ANN.

- Iterasi metode BP dalam pencarian konfigurasi bobot koneksi ANN dapat terjebak dalam *local optimum*. Local optimum adalah kondisi saat ditemukannya solusi terbaik terhadap iterasi sebelumnya, tetapi bukan *global optimum* (Gambar 2.4). Konfigurasi ANN local optimum ini dapat diterima bila posisinya berdekatan dengan global optimum, akan tetapi hal ini sangat jarang terjadi.



Gambar 2.4 Ilustrasi Global dan Local Optimum (Olson, 2014)

2.2.2. Parameter dan Kinerja ANN

Kendala yang harus diselesaikan terlebih dahulu sebelum proses pelatihan dilakukan adalah menentukan beberapa parameter awal (Ortiz, et al., 2011). Sampai saat ini belum ada satu aturan baku yang bisa digunakan untuk menentukan parameter-parameter awal tersebut. Dalam prakteknya, parameter-parameter itu ditentukan secara subyektif sesuai dengan kebutuhan dan kondisi data latih yang digunakan serta berdasarkan pengalaman. Berikut beberapa parameter yang perlu dicermati sebelum proses pelatihan dimulai, antara lain :

- Arsitektur ANN

Tidak adanya aturan baku tentang arsitektur ANN mengakibatkan perancangan arsitektur ANN dilakukan secara *trial and error* (coba-coba). Kompleksitas permasalahan dapat digunakan sebagai salah satu pertimbangan dalam merancang arsitektur ANN. Bila ukuran jaringan terlalu kecil, ANN tidak dapat belajar dengan baik. Bila ukurannya terlalu besar, akan mengakibatkan over-fitting. Penentuan jumlah *hidden layer* dan jumlah neuron dalam setiap hidden layer perlu dilakukan secara hati-hati.

- Algoritma dan Parameter Pelatihan

Sampai saat ini, ada beberapa algoritma pelatihan yang bisa digunakan, salah satunya adalah Back Propagation (BP). Walaupun BP adalah salah satu algoritma yang paling banyak dipakai, akan tetapi BP tidak bisa menjamin kualitas terbaik untuk semua jenis permasalahan dan dataset. Pemilihan satu algoritma tertentu bisa saja memberikan hasil terbaik untuk sebagian dataset tertentu, akan tetapi berperilaku sebaliknya untuk bagian dataset lainnya. Pemilihan nilai konstanta (parameter) juga dilakukan secara subyektif. Sampai saat ini belum ada satu aturan baku yang bisa digunakan untuk menentukan nilai *learning rate* dan *momentum* dalam proses pelatihan.

- Data Latih

Kuantitas dan kualitas yang digunakan dalam proses pelatihan berperan sangat penting dalam menentukan kualitas output ANN. Kuantitas data latih yang cukup besar, biasanya digunakan untuk menjamin kualitas proses pelatihan, akan tetapi berakibat pada waktu komputasi yang lama. Untuk tetap menjaga kualitas proses pelatihan dan juga mengurangi waktu komputasi, pemilihan data latih perlu dilakukan dengan tetap memperhatikan keberagamannya. *Data pre-processing* dapat dilakukan terlebih dahulu sebelum melakukan proses pelatihan untuk menjamin kualitas proses pelatihan.

- Sub-dataset Latih (*Training Set*)

Untuk suatu kasus tertentu, dataset keseluruhan berukuran sangat besar, sehingga tidak memungkinkan untuk menggunakan keseluruhan dataset tersebut dan hanya menggunakan sebagian saja untuk proses pelatihan. Pemilihan sub-dataset itu harus dilakukan sedemikian hingga ANN tetap memberikan output yang benar walaupun diuji dengan data yang tidak terdapat dalam sub-dataset tersebut.

- Relasi Proses

ANN dapat dikategorikan sebagai pemodelan yang bersifat black-box. Hasil akhir (output) yang akan didapatkan melalui ANN cukup sulit dianalisis dan dikaitkan dengan input vektor.

Park memberikan beberapa petunjuk praktis yang bisa digunakan dalam menentukan parameter-parameter awal dalam ANN pada

Tabel 2.3 (Park, 2011).

Tabel 2.3 Formula praktis parameter ANN

Formula	Keterangan
$h_n = 2i_n + 1$	Hetch-Nelson menggunakan teori Kolmogorov untuk mendapatkan jumlah hidden node (Hecht-Nielsen, 1987)
$h_n = \frac{(i_n + o_n)}{2}$	Lawrence, et al. menyarankan jumlah hidden node (Lawrence & Patterson, 1993)
$h_n = i_n \log_2 P$	Marchandani, et al. menyarankan jumlah hidden node (Mirchandani & Wei, 1989)
$2T_n \leq N_d \leq 10T_n$ $T_n = i_n + h_n + o_n$	Lawrence, et al. menyarankan jumlah data latih (Lawrence & Patterson, 1993)

Keterangan:

h_n = jumlah hidden node

i_n = jumlah input node

o_n = jumlah output node

T_d = jumlah seluruh node, yaitu: input, hidden dan output

N_d = jumlah data latih

Formulasi diatas adalah persamaan empirik yang didapatkan dari hasil pengujian dan bersifat saran. Penyesuaian dan perubahan masih diperlukan untuk menyesuaikan dengan kebutuhan dan kondisi data latih.

2.2.3. Kesalahan Prediksi

Untuk setiap nilai hasil prediksi \hat{y} terhadap y , memungkinkan terdapat perbedaan sebesar $y - \hat{y}$, perbedaan nilai ini disebut sebagai residual, kesalahan estimasi atau kesalahan prediksi (Larose, 2005). Proses prediksi menggunakan ANN diupayakan sedemikian rupa untuk meminimalkan kesalahan prediksi. Untuk mengetahui kinerja prediksi ANN, digunakan rumus Median Absolute Percentage Error (MdAPE) sebagai berikut (Shcherbakov, et al., 2013) :

$$\text{MdAPE} = \text{median}_{i=1,n} \left(\left| \frac{y - \hat{y}}{y} \right| \cdot 100 \right) \quad (2.8)$$

Keterangan:

y = atribut output dari dataset

\hat{y} = hasil prediksi

2.2.4. Normalisasi Dataset dan Denormalisasi Hasil Prediksi

Bila dataset memiliki output y yang bernilai sangat kecil mendekati nol, maka nilai MdAPE menjadi sangat besar, melebihi 100%. Hal ini mengakibatkan, performa model prediksi ANN menjadi sangat buruk. Selain itu, bila masing-masing atribut dalam dataset memiliki nilai yang sangat jauh berbeda amplitudonya, maka model prediksi ANN juga akan menurun akurasi. Untuk menghindari hal tersebut, dataset yang akan digunakan dalam perlu dinormalisasi terlebih dahulu sehingga setiap atribut dalam dataset memiliki amplitudo nilai yang seragam. Metode normalisasi yang akan digunakan adalah metode MinMax yang dirumuskan sebagai berikut (Ogasawara, et al., 2010) :

$$S = S_{low} + (S_{high} - S_{low}) \times \frac{A - A_{min}}{A_{max} - A_{min}} \quad (2.9)$$

Untuk mengembalikan hasil prediksi ke skala awal, dilakukan dengan rumus denormalisasi sebagai berikut :

$$A = A_{min} + (A_{max} - A_{min}) \times \frac{S - S_{low}}{S_{high} - S_{low}} \quad (2.10)$$

Keterangan:

- A = data dalam skala awal
- A_{min} = nilai terbesar dalam data A
- A_{max} = nilai terkecil dalam data A
- S = data dalam skala MinMax
- S_{high} = batas atas data S
- S_{low} = batas bawah data S

Umumnya, nilai $S_{low} = 0$ dan $S_{high} = 1$, sehingga hasil normalisasi data akan menghasilkan rentang nilai antara 0 s/d 1.

2.2.5. Fungsi Aktivasi

Proses pelatihan ANN dilakukan dalam beberapa kali sesi iterasi sampai mencapai konvergensi. Setiap sesi iterasi pelatihan dapat menghasilkan bobot link yang sangat besar nilainya, sehingga bila nilai-nilai ini dimasukkan dalam fungsi aktivasi sigmoid akan menghasilkan nilai yang konstan (Larose, 2005). Untuk mengatasi hal tersebut, Nair dan Hinton (Nair & Hinton, 2010) mengusulkan penggunaan fungsi aktivasi Rectified Linear Units (ReLU) yang dirumuskan pada persamaan 2.11.

$$y = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.11)$$

2.2.6. Normalisasi Model Prediksi

Untuk meningkatkan kinerja fungsi aktivasi yang telah dijelaskan di Bab 2.2.5, maka model prediksi ANN perlu untuk dinormalisasi nilai bobot setiap link. Normalisasi model prediksi dilakukan dengan algoritma yang diusulkan oleh Diehl, et.al. pada Gambar 2.5 (Diehl, et al., 2015).

```

1 | for layer in layers:
2 |     max_pos_input = 0
3 |     # Find maximum input for this layer
4 |     for neuron in layer.neurons:
5 |         input_sum = 0
6 |         for input_wt in neuron.input_wts:
7 |             input_sum += max(0, input_wt)
8 |         max_pos_input = max(max_pos_input, input_sum)
9 |     # Rescale all weights
10 |    for neuron in layer.neurons:
11 |        for input_wt in neuron.input_wts:
12 |            input_wt = input_wt / max_pos_input

```

Gambar 2.5 Normalisasi model prediksi

2.2.7. Validasi Silang

Untuk meningkatkan kualitas model prediksi dengan jumlah data latih dan uji yang sedikit, Bishop menyarankan memaksimalkan penggunaan keseluruhan data untuk pelatihan dan pengujian dengan cara pembagian data dalam beberapa partisi (Bishop, 2006). Setiap partisi data digunakan secara bergantian dalam proses pelatihan dan pengujian. Metode ini sering disebut sebagai metode K-fold atau S-fold. Untuk lebih jelasnya dapat diamati ilustrasi pada Gambar 2.6.

1	2	3	4	run 1
1	2	3	4	run 2
1	2	3	4	run 3
1	2	3	4	run 4

Gambar 2.6 Ilustrasi partisi data latih dan uji dengan metode K-fold

Dalam gambar diatas, diuraikan implementasi metode K-fold dengan partisi data sebanyak 4 kelompok. Dalam setiap iterasi pelatihan dan pengujian, 1 partisi/kelompok data digunakan untuk pengujian dan partisi lain digunakan untuk pelatihan. Di iterasi berikutnya, partisi yang lain digunakan sebagai data pengujian secara bergantian terus menerus sampai proses pelatihan berakhir.

2.3. Genetic Algorithm (GA)

Sejak diusulkan pertama kali tahun 1975 oleh John Henry Holland, Genetic Algorithm (GA) telah digunakan secara luas diberbagai bidang ilmu pengetahuan terutama untuk menyelesaikan permasalahan optimasi (Holland, 2012). Sampai saat ini, GA telah berkembang sangat pesat dan telah menghasilkan beberapa algoritma turunan diantaranya Differential Evolutionary, Evolution Strategies, Evolutionary Programming, Genetic Programming, Grouping Genetic Algorithm dan Interactive Evolutionary Algorithm. Selain kemudahan implementasi, metode optimasi GA juga memiliki kelebihan lain, yaitu:

- Cakupan pencarian yang lebar.

Cakupan pencarian solusi yang cukup lebar memungkinkan GA menemukan solusi *Global Optimum* secara mudah. Keragaman (*diversity*) populasi awal menentukan cakupan pencarian solusi, sehingga biasanya populasi awal disusun dari solusi-solusi yang cukup jauh jaraknya.

- Kecepatan Tinggi.

Bila dibandingkan dengan beberapa metode optimasi lain, metode optimasi GA dapat menemukan solusi *Global Optimum* dalam waktu relatif singkat.

- Dimensi Tinggi.

Walaupun pada awalnya diusulkan untuk masalah optimasi yang sederhana, pada perkembangannya GA juga mampu menyelesaikan masalah optimasi berdimensi tinggi.

- Stabilitas hasil optimasi.

Sama seperti metode optimasi heuristik lainnya, metode optimasi GA juga membutuhkan penentuan beberapa konstanta di awal iterasi. Metode optimasi GA dapat saja menunjukkan hasil yang sama, walaupun iterasi dimulai dengan konstanta yang berbeda.

- Banyaknya penelitian terdahulu.

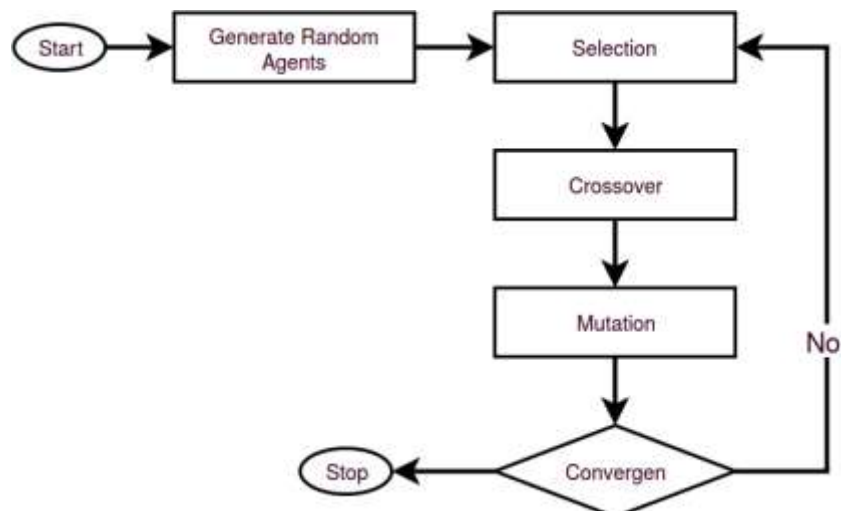
Perkembangan implentasi GA demikian pesat di berbagai bidang, sehingga ada cukup banyak referensi yang bisa digunakan bahan acuan.

Implementasi metode optimasi GA dimulai dengan membentuk populasi awal berisi sejumlah gen (beberapa referensi menyebut sebagai kromosom, *phenotype*, individu atau *agent*). Masing-masing gen mewakili sebuah solusi yang telah disandikan (*encoded solution*). Sampai saat ini ada 3 cara untuk melakukan encoding gen di GA, yaitu : gen string, gen biner dan gen nilai langsung (*direct-encoded value*). Gambar 2.7 adalah ilustrasi hasil proses *encoding* gen di metode GA.

Gen String	A	S	D	Q	W	...	C
Gen Biner	0	1	0	1	1	...	0
Gen Nilai Langsung	0.9	0.1	0.77	1.2	0.2	...	1.1

Gambar 2.7 Bentuk gen GA hasil proses *encoding*

Selanjutnya, proses perhitungan dilanjutkan dengan melakukan iterasi perhitungan yang tersusun atas operasi seleksi, operasi rekombinasi (*crossover*) dan operasi mutasi. Setiap iterasi perhitungan disebut sebagai sebuah generasi. Proses iterasi berhenti bila telah mencapai kondisi tertentu, yaitu pada jumlah iterasi tertentu atau kondisi konvergen (mencapai error yang kecil). Untuk lebih jelasnya dapat diamati pada Gambar 2.8.



Gambar 2.8 Iterasi metode optimasi Genetic Algorithm

Setiap gen dalam GA tersusun atas beberapa allele. Jumlah allele dalam setiap gen adalah jumlah variabel dari *objective function* dalam sebuah permasalahan optimasi. Tingkat kesesuaian (*fitness*) masing-masing gen diuji dengan nilai yang didapatkan dari *fitness function*. Dalam metode optimasi GA, terdapat beberapa konstanta parameter yang perlu ditentukan lebih dahulu (Tabel 2.4).

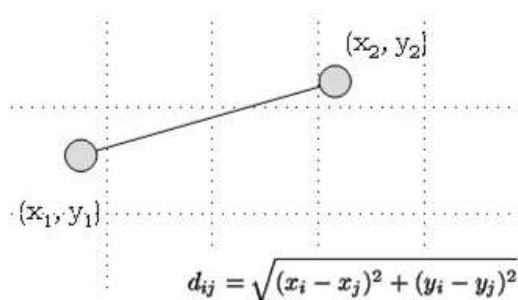
Tabel 2.4 Konstanta parameter GA

No	Konstanta	Keterangan
1	N	Jumlah gen dalam populasi.
2	P_{cr}	Peluang rekombinasi (<i>crossover</i>), interval [0,1]
3	P_m	Peluang mutasi (<i>mutation</i>), interval [0,1]

2.3.1. Pembangkitan Populasi Acak

Metode optimasi GA diawali dengan pembangkitan populasi acak, yaitu membuat beberapa gen dengan nilai allele acak. Untuk gen bertipe string dan biner, populasi awal yang dibangkitkan akan selalu menghasilkan gen-gen yang benar-benar unik, sehingga tingkat keragaman (*diversity*) populasi sangatlah baik. Seperti telah dijelaskan di sub bab terdahulu, cakupan pencarian solusi GA ditentukan oleh tingkat keragaman sebuah populasi.

Pada gen bertipe nilai langsung (misalnya: bilangan riil), setiap individu yang dibangkitkan untuk populasi awal, sangat memungkinkan untuk memiliki tingkat kemiripan (*similarity*) yang tinggi, sehingga mengurangi diversity populasi dan cakupan pencarian solusi GA. Untuk gen bertipe bilangan riil, tingkat kemiripan antara satu gen dengan gen lain dapat diukur menggunakan rumus jarak Euclidean. Jarak euclidean yang cukup jauh (misalnya $\varepsilon > 10^{-6}$ atau bahkan lebih besar lagi) menjamin tingkat diversity populasi awal yang lebih baik (Gupta & Ghafir, 2012). Gambar 2.9 adalah ilustrasi perhitungan jarak Euclidean.



Gambar 2.9 Gambaran Jarak Euclidean

Setelah populasi acak terbentuk, masing-masing gen dihitung nilai fitnessnya berdasarkan *fitness function*. Pembangkitan setiap gen secara acak dilakukan dalam iterasi sampai jumlah populasi awal terpenuhi.

2.3.2. Operasi Seleksi N-Tournament

Operasi seleksi Tournament dimulai dengan memilih sejumlah **2** gen secara acak (*pair-wise*) kemudian membandingkan *fitness value* setiap gen dan memilih salah satu yang terbaik (disesuaikan dengan *objective function*). Gen terbaik yang terpilih kemudian diletakkan dalam sebuah populasi jodoh (*mate pool*). Proses diatas diulangi sebanyak **N** kali sampai jumlah gen dalam populasi jodoh memenuhi kebutuhan operasi GA selanjutnya (Deb, 2000).

Bila $N=2$, maka metode seleksi ini disebut sebagai *binary tournament*, yaitu memilih secara acak 2 gen dari populasi kemudian mencari gen terbaik untuk diletakkan di populasi jodoh. Beberapa referensi menganjurkan nilai $N>2$ untuk mengurangi tekanan pencarian hanya pada gen-gen terbaik saja (Gupta & Ghafir, 2012).

2.3.3. Operasi Rekombinasi BLX- α

Operasi rekombinasi Blend Alpha Crossover (BLX- α) adalah operasi rekombinasi nilai allele antara 2 induk dengan langkah-langkah sebagai berikut :

1. Pilih secara acak 2 induk dari populasi, x^1 dan x^2
2. Nilai allele *offspring* x^c pada posisi ke- i , x_i^c dipilih secara random dalam interval yang terdistribusi secara uniform $[X_i^1, X_i^2]$, selanjutnya :

$$X_i^1 = \min(x_i^1, x_i^2) - \alpha * d_i \quad (2.12)$$

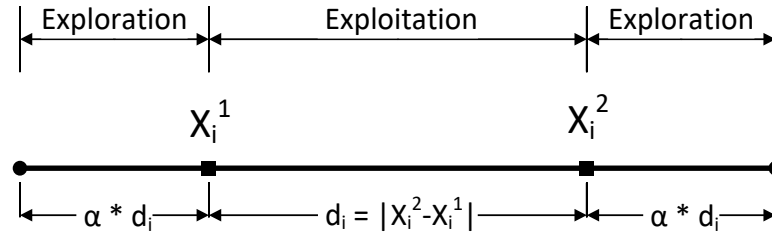
$$X_i^2 = \max(x_i^1, x_i^2) - \alpha * d_i \quad (2.13)$$

$$d_i = |x_i^1 - x_i^2| \quad (2.14)$$

x_i^1 dan x_i^2 adalah allele ke- i dari induk x^1 dan x^2 , α adalah parameter positif dalam interval $[0,1]$. Picek menggunakan $\alpha = 0.5$ (Picek, et al., 2013), Takashi menggunakan $\alpha = 0.366$ (Takahashi & Kita, 2001).

Aspek penting dalam operasi rekombinasi adalah aspek eksploitasi dan eksplorasi. Bila parameter α yang digunakan adalah $\alpha = 0$ maka operasi rekombinasi yang dilakukan lebih bersifat eksploitatif, yaitu mencari nilai allele baru dari dalam interval nilai allele induk.

Semakin besar nilai parameter α yang digunakan (biasanya mendekati 1, $\alpha < 1$), maka semakin eksploratif yaitu mencari nilai allele baru di luar nilai range allele induk sehingga memperluas cakupan (*coverage*) pencarian solusi. Untuk lebih jelasnya dapat diamati pada Gambar 2.10.



Gambar 2.10 Eksploitasi dan eksplorasi BLX- α

2.3.4. Operasi Rekombinasi Multi Induk

Beberapa studi terakhir menunjukkan bahwa rekombinasi menggunakan beberapa induk (lebih dari dua induk) mampu meningkatkan kinerja metode optimasi GA, antara lain: Center of Mass Crossover (CMX), Multi-parent Feature-wise Crossover (MFX), Seed Crossover (SX) dan Multi-modal Distribution Crossover (MMDX) (Tsutsui & Ghosh, 1998) (Ariyarit & Kanazaki, 2015). Dari hasil uji coba yang telah dilakukan di studi-studi terdahulu, didapatkan beberapa kesimpulan, antara lain: mampu menjaga *diversity* populasi tetap tinggi dan menyelesaikan permasalahan yang rumit (multi-modal dan epistasis). Dalam studi-studi tersebut, jumlah induk yang digunakan dalam operasi rekombinasi bervariasi, mulai dari 2 induk sampai 16 induk.

2.3.5. Operasi Mutasi Bilangan Riil

Yoon, Yourim dan Kim, Yong-Hyuk mengusulkan *Fine Mutation* (FM), sebuah metode operasi mutasi yang diinspirasi oleh metode mutasi Gaussian (Yoon & Kim, 2012). Fine Mutation dirumuskan sebagai berikut :

$$z_i = z_i + Nd(0, |x_i - y_i|) \quad (2.15)$$

Keterangan:

Nd = distribusi normal dengan standard deviasi $|x_i - y_i|$

x_i, y_i = allele gen induk yang ke-i

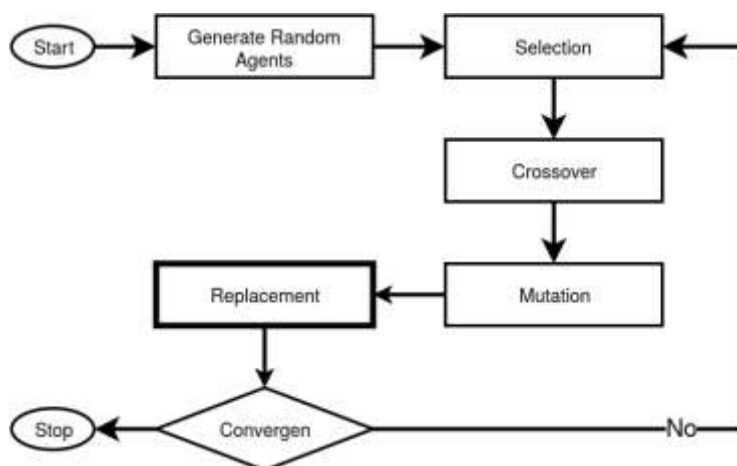
z_i = allele gen offspring z yang ke-i

Operasi mutasi GA pada umumnya hanya menambahkan nilai absolut yang nilainya sangat kecil dengan peluang mutasi yang juga sangat kecil dibandingkan dengan peluang rekombinasi. FM menawarkan konsep baru dengan pemanfaatan nilai-nilai gen induk untuk mendapatkan besaran nilai mutasi. Besaran nilai mutasi metode FM secara konseptual lebih baik karena dapat beradaptasi dengan tingkat konvergensi dalam setiap generasi (iterasi) GA. Dalam konsep FM, konvergensi setiap generasi terwakili dalam setiap induk yang terpilih dalam operasi seleksi.

2.3.6. Steady State GA

Steady State Genetic Algorithm (SSGA) adalah salah satu turunan metode GA yang jumlah populasi setiap generasi (iterasi) adalah sama. SSGA pertama kali diusulkan oleh Darrell Whitley tahun 1989 (Lozano, et al., 2008). SSGA memiliki jumlah populasi yang relatif kecil bila dibandingkan dengan GA konvensional, sehingga kebutuhan memori (RAM) SSGA relatif lebih sedikit. SSGA dirancang sebagai metode GA yang lebih ringkas dan gegas namun memiliki performa berimbang GA konvensional.

Kompleksitas komputasi antara GA konvensional dan SSGA tidak jauh berbeda karena kedua metode sama-sama melakukan operasi seleksi, rekombinasi dan mutasi. Perbedaan utamanya adalah pada jumlah populasi yang tetap (*steady*) dan bertambahnya operasi penggantian. Gambar 2.11 adalah ilustrasi alur SSGA.



Gambar 2.11 Pemodelan Steady State GA

2.3.7. Operasi Penggantian dalam SSGA

Salah satu perbedaan SSGA dengan GA konvensional adalah adanya operasi penggantian (*replacement*) untuk memastikan bahwa jumlah populasi tetap sama. Jumlah gen pengganti hasil operasi rekombinasi (*crossover*) dan mutasi sama jumlahnya dengan jumlah gen yang akan digantikan. Teknik penggantian yang paling banyak digunakan adalah Worst Replacement (WR) (Lozano, et al., 2008). WR menggantikan gen dengan *fitness function* terburuk dengan gen dengan *fitness function* yang lebih baik.

Teknik WR berakibat pada bertambahnya tekanan pada operasi seleksi sehingga hanya gen terbaik saja yang akan terpilih dalam operasi seleksi. Dalam beberapa kasus, teknik ini mengakibatkan SSGA mencapai konvergen sangat cepat tetapi juga dapat terjebak dalam *premature convergence*.

2.4. Hibridasi ANN-GA

Sejak tahun 1989, metode optimasi GA sudah digunakan dalam proses pelatihan ANN (Montana & Davis, 1989). Sampai saat ini, sudah cukup banyak studi dan penelitian yang dilakukan tentang penggunaan metode optimasi GA dalam pelatihan ANN menggantikan BP, antara lain: prediksi hasil pengelasan (Zhang, et al., 2015), transistor *modelling* (Jarndal, 2015), *viral production* (Takahashi, et al., 2015), dan masih banyak lagi. Kelebihan-kelebihan yang dimiliki oleh GA, membuat metode optimasi GA menjadi salah satu pilihan dalam melatih ANN. Kelebihan metode optimasi GA telah dijelaskan dalam bagian awal Bab 2.3.

Dari studi literatur diatas, konfigurasi ANN, yaitu bobot masing-masing koneksi antar node, di-*encode* menjadi sebuah gen. Setiap nilai bobot koneksi digunakan sebagai sebuah nilai allele. Nilai bobot koneksi bertipe bilangan riil, sehingga gen dalam ANN-GA juga bertipe bilangan riil. Jumlah keseluruhan koneksi dalam ANN, mulai dari input layer, hidden layer sampai output layer adalah jumlah allele dalam setiap gen di GA.

Tujuan (*objective*) hibridasi ANN-GA adalah melakukan optimasi global terhadap konfigurasi ANN. Hibridasi ANN-GA dapat menyelesaikan masalah-masalah yang dihadapi metode pelatihan Back Propagation, yaitu menghindari kondisi *local optimum* dan *premature convergence*. Hal ini dimungkinkan karena pencarian solusi yang dilakukan GA bersifat global dan memiliki cakupan pencarian solusi yang lebih luas.

Dari uraian diatas dapat disimpulkan bahwa tujuan penggunaan metode optimasi GA dalam pelatihan ANN adalah meningkatkan kinerja ANN, yaitu memaksimalkan akurasi ANN dan meminimalkan kesalahan (*error*) prediksi ANN. Hal ini sangat dimungkinkan karena hibridasi ANN-GA mampu menemukan kombinasi bobot koneksi antar node ANN yang menghasilkan kesalahan prediksi terkecil (Montana & Davis, 1989). *Fitness function* masing-masing gen dirumuskan sebagai kesalahan prediksi menggunakan rumus MdAPE pada Bab 2.2.3 diatas. Pencarian konfigurasi optimum dilakukan dalam beberapa iterasi.

BAB 3

METODOLOGI PENELITIAN

Tahapan-tahapan yang dilalui pada penelitian ini meliputi (1) Studi Literatur, (2) Desain Model Sistem, (3) Pembuatan Perangkat Lunak, (4) Pengujian Sistem, dan (5) Analisa Hasil Pengujian. Gambar 3.1 menunjukkan tahapan-tahapan tersebut secara berurut.



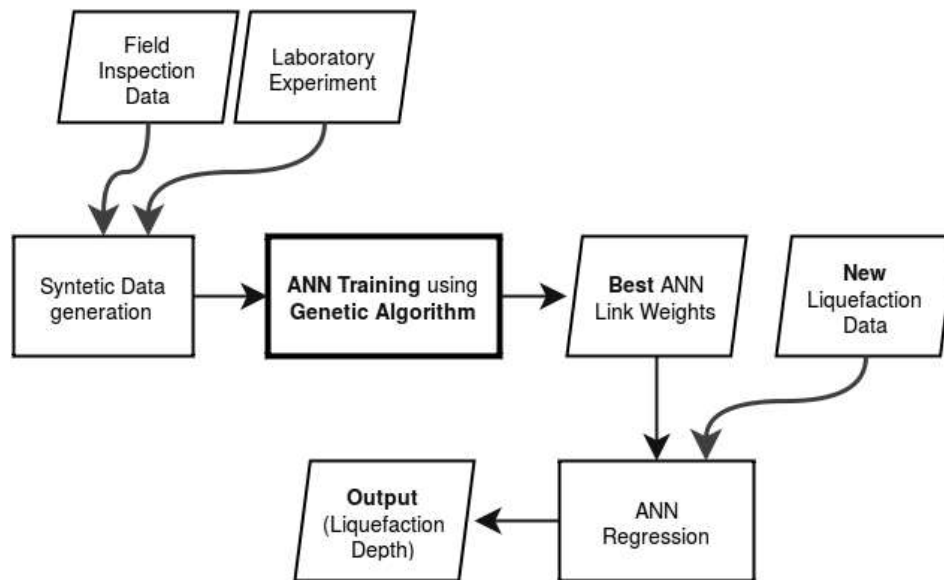
Gambar 3.1 Tahapan Penelitian

3.1. Studi Literatur

Studi literatur dalam penelitian ini dilakukan dengan mengumpulkan dan memahami berbagai publikasi hasil penelitian terbaru dan beberapa buku teks tentang Artificial Neural Network (ANN) dan Genetic Algorithm (GA). Untuk Artificial Neural Network, studi literatur berfokus pada proses pelatihan jaringan. Sementara untuk Genetic Algorithm, studi literatur yang dilakukan berfokus pada pengembangan operator-operator dalam GA, mulai dari operator seleksi, rekombinasi (*crossover*) dan mutasi.

3.2. Desain Model Sistem

Metodologi yang diusulkan dalam penelitian adalah penggunaan metode optimasi GA untuk melakukan pelatihan terhadap ANN. Untuk lebih jelasnya dapat diamati pada Gambar 3.2.



Gambar 3.2 Alur Penelitian

3.2.1. Pembangkitan Dataset

Data yang akan digunakan dalam penelitian ini, diambil dari beberapa penelitian terdahulu (Nataraja & Gill, 1983) (Jeng, 1997) (Cha, et al., 2006) (Cha, et al., 2009) (Liao, et al., 2015) (Liao, et al., 2015). Data terbagi dalam dua kelompok, yaitu data gelombang dan data karakteristik tanah (soil properties). Data tersebut memiliki beberapa atribut, antara lain :

1. Konstanta *relative density* (D_r , tanpa dimensi)
2. Konstanta *earth pressure at rest* (K_0 , tanpa dimensi)
3. Periode gelombang (T, detik)
4. Kedalaman air (d, meter)
5. Tinggi gelombang (H, meter)
6. Kedalaman Liquefaction (z, meter, **output**)

Output prediksi adalah variabel bertipe real yang menunjukkan terjadinya fenomena liquefaction pada satu kedalaman tanah tertentu.

Persiapan data yang dilakukan adalah pembangkitan data variasi, yaitu membangkitkan variasi data input berdasarkan data yang didapatkan dari penelitian terdahulu. Dari setiap variasi data yang telah diperoleh, kemudian dilanjutkan pembangkitan data sintetis yaitu data output (kedalaman maksimum *liquefaction*, *z*) melalui perhitungan *liquefaction* menggunakan persamaan yang diusulkan oleh Ishihara seperti telah dijelaskan pada Bab 2.1 persamaan (2.6) (Ishihara & Yamazaki, 1984). Pada Tabel 3.1 adalah contoh dataset yang digunakan dalam penelitian ini.

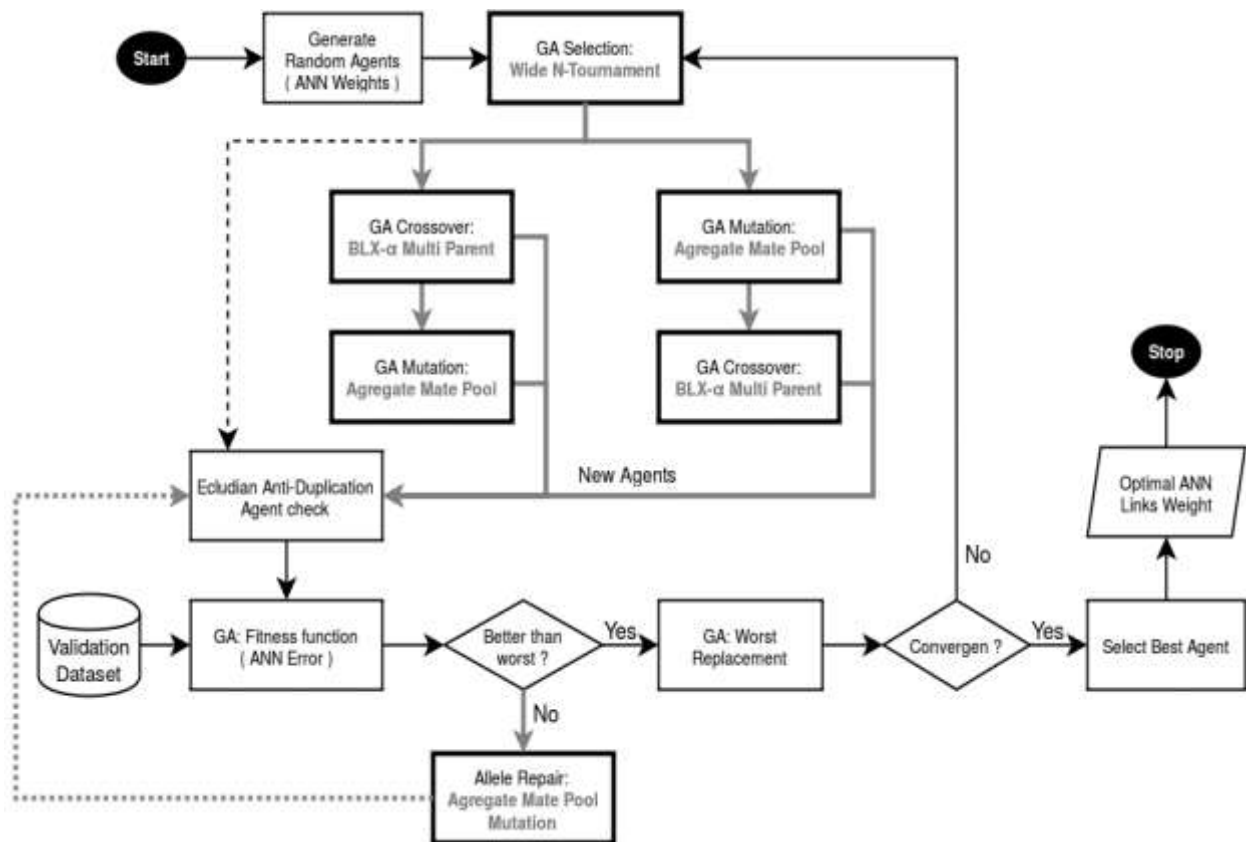
Tabel 3.1 Contoh sebagian dataset Liquefaction

Input					Output
CSR			CRR		z
T	d	H	K0	Dr	
15	30	25	0.2	0.65	212.94
14	10	5	0.2	0.65	103.33
16	30	20	0.18	0.65	220.53
14	20	10	0.16	0.75	146.35
13	10	5	0.2	0.75	93.82
12	50	15	0.16	0.75	151.86
16	10	5	0.18	0.65	116.92
13	50	20	0.2	0.85	176.42
15	30	20	0.16	0.85	196.81
13	70	20	0.16	0.85	174.77
16	30	25	0.16	0.75	224.96
16	70	10	0.2	0.85	205.81
13	70	5	0.16	0.85	119.83
16	70	5	0.16	0.65	185.86
13	30	10	0.16	0.65	153.75
15	70	20	0.2	0.75	227.09
14	20	25	0.2	0.75	171.28
13	10	25	0.2	0.65	128.31
15	10	15	0.18	0.85	129.56
14	10	5	0.18	0.65	103.95

3.2.2. Pelatihan ANN menggunakan GA

Proses pelatihan ANN menggunakan metode optimasi GA adalah optimasi bobot setiap *link* (koneksi) di ANN menggunakan teknik hibridasi ANN-GA seperti yang telah dijelaskan pada Bab 2.4. Proses evolusi (iterasi) GA akan berhenti bila konfigurasi ANN yang terbaik telah

ditemukan, yaitu konfigurasi ANN yang memiliki kesalahan prediksi MdAPE terendah. Penjelasan tentang konvergensi GA dijelaskan kemudian di Bab 3.2.2.8.



Gambar 3.3 Alur pelatihan ANN menggunakan metode optimasi GA

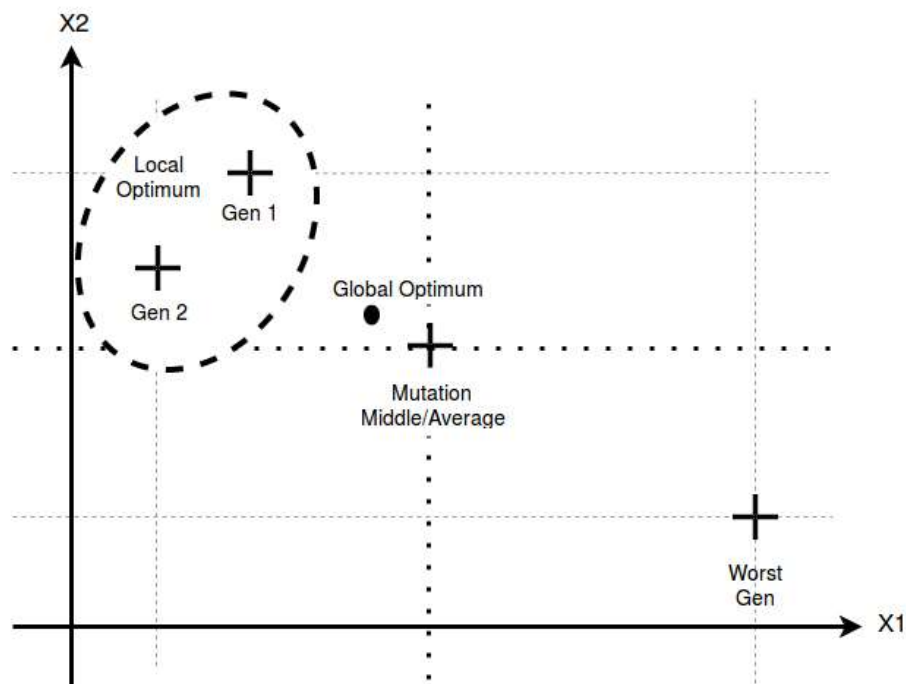
Diagram alur pada Gambar 3.3 adalah usulan modifikasi metode optimasi GA dalam proses pelatihan ANN. Modifikasi metode optimasi GA tersebut kemudian disebut sebagai Wide GA (WGA). Teknik hibridasi antara ANN dan WGA kemudian disebut sebagai ANN-WGA. Usulan metode WGA terdiri dari Operasi Seleksi Wide Tournament, Operasi Rekombinasi BLX- α Multi-Induk, Operasi Mutasi Agregate Mate Pool dan Operasi Rekombinasi-Mutasi Langsung. Keempat usulan operasi baru tersebut dan beberapa operasi tambahan akan dijelaskan dalam beberapa sub bab beriku ini.

3.2.2.1. Operasi Seleksi Wide Tournament

Seperti telah dijelaskan dalam Bab 2.3.2, pada umumnya, operasi seleksi GA memberikan tekanan cukup besar untuk memilih gen-gen terbaik saja, misalnya metode seleksi *elitist* (Gupta & Ghafir, 2012). Tekanan yang terlalu besar dapat mengakibatkan *diversity* populasi menurun, menyempitnya cakupan ruang pencarian dan proses pencarian solusi dapat

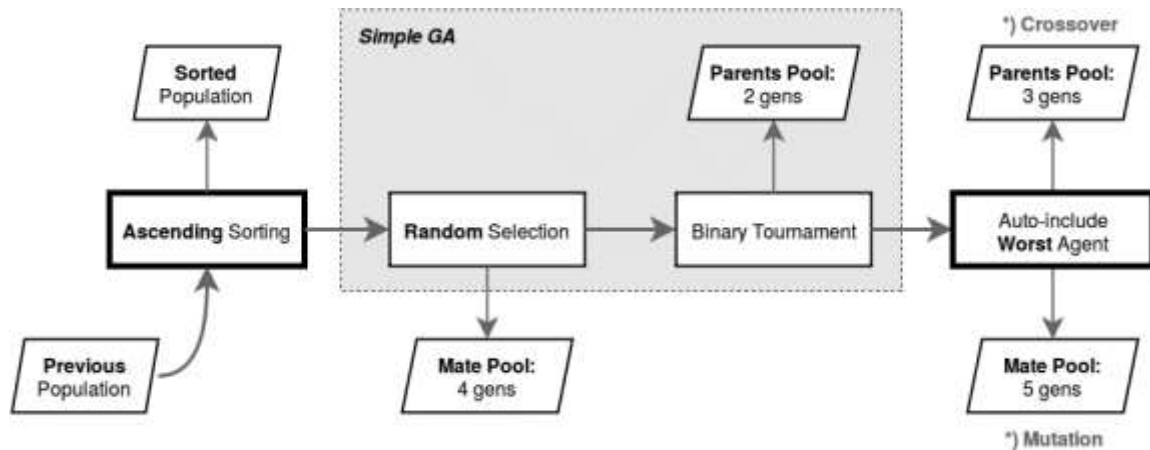
terjebak pada kondisi *local optimum* dan *premature convergence*. Dalam proses Tournament, pemilihan gen secara acak juga dapat memperbesar tekanan pemilihan gen-gen terbaik saja bila pemilihan gen tersebut hanya melibatkan 2 gen saja (Deb, 2000).

Berdasarkan hal-hal diatas, diusulkan sebuah operasi seleksi Wide Tournament, yaitu operasi seleksi yang selalu menyertakan gen terburuk dalam populasi jodoh dan proses Tournament melibatkan lebih banyak gen (lebih dari 2 gen). Operasi seleksi Wide Tournament bertujuan untuk memperluas cakupan (*coverage*) pencarian kandidat solusi sehingga GA dapat mengeksplorasi ruang solusi yang lebih luas. Semakin luas ruang solusi yang dapat dieksplorasi maka semakin mudah bagi GA untuk menemukan solusi optimum global. Dalam penelitian ini, proses Tournament akan memilih 4 gen secara acak dan menambahkan 1 gen terburuk dari populasi jodoh. Untuk lebih jelasnya dapat diamati pada Gambar 3.4.



Gambar 3.4 Konsep Operasi Seleksi Wide Tournament

Sebelum melakukan proses seleksi, dilakukan terlebih dahulu proses pengurutan menaik (*ascending sorting*) gen berdasarkan nilai *fitness function*-nya, dalam hal ini nilai MdAPE setiap gen. Gen terbaik pada urutan pertama dan gen terburuk pada urutan terakhir. Untuk lebih jelasnya dapat diikuti bagan alur di Gambar 3.5.



Gambar 3.5 Alur Seleksi Wide Tournament

Bila populasi gen berisi M gen, maka pencarian 4 gen acak akan dilakukan terhadap M-1 gen karena gen terakhir (yang juga gen terburuk) secara otomatis terpilih sebagai salah satu induk untuk proses rekombinasi. 4 gen yang terpilih secara acak kemudian dimasukkan kedalam populasi jodoh (*mate pool*). Proses turnamen dilakukan 2 kali untuk mencari 2 gen kandidat induk. Kandidat induk yang terpilih kemudian dimasukkan dalam populasi induk (*parent pool*). Proses dilanjutkan dengan memasukkan gen terburuk dalam populasi induk dan populasi jodoh. Populasi induk akan digunakan dalam operasi rekombinasi (*crossover*) dan populasi jodoh akan digunakan dalam operasi mutasi.

Operasi seleksi Wide Tournament dapat mempercepat proses pencarian di awal iterasi bila dipadukan dengan operasi rekombinasi BLX- α Multi-Induk yang akan dijelaskan di bab selanjutnya.

3.2.2.2. Operasi Rekombinasi BLX- α Multi-Induk

Seperti telah diuraikan dalam Bab 2.3.4 tentang operasi rekombinasi multi-induk, keberadaan induk tambahan dalam proses rekombinasi dapat mengurangi tekanan dalam proses seleksi, meningkatkan keragaman (*diversity*) populasi dan menghindari *incest* (rekombinasi antar gen yang mirip) (Tsutsui & Ghosh, 1998) (Ariyarit & Kanazaki, 2015). Operasi rekombinasi BLX- α Multi-Induk menggunakan gen terburuk sebagai tambahan induk ketiga. Proses pemilihan gen terburuk dilakukan dalam proses seleksi Wide Tournament yang telah dijelaskan di bab terdahulu, Bab 2.3.2. Penambahan gen terburuk ini bertujuan untuk memberikan arah baru eksplorasi ruang solusi sehingga upaya eksplorasi menjadi lebih luas cakupannya (*coverage*).

Operasi rekombinasi BLX- α Multi-Induk tidak membangkitkan nilai acak dari sebuah interval yang didapatkan dari nilai-nilai allele gen induk, tetapi menggunakan secara langsung nilai-nilai agregat, yaitu nilai batas atas (*upper*), nilai batas bawah (*lower*), nilai tengah (*middle*) dan nilai rata-rata (*average*). Setiap operasi rekombinasi akan menghasilkan beberapa turunan gen (*offspring*), yaitu Z^{upper} , Z^{lower} , Z^{middle} dan $Z^{average}$. Proses pembuatan gen turunan (*offspring*) dimulai dengan membuat *clone* salah satu induk kemudian melakukan perubahan nilai allele offspring. Proses *cloning* gen induk dan perubahan nilai allele diulang sebanyak kombinasi antara induk yang terlibat dalam proses rekombinasi dan jenis nilai agregat (*upper*, *lower*, *middle*, *average*).

Bila dipilih secara acak sebuah allele urutan ke- i , jumlah induk dalam proses rekombinasi adalah m dan gen induk terlibat adalah $X^1, X^2, X^3, \dots, X^m$, maka dapat dihitung :

$$y_i^{min} = \min(x_i^1, x_i^2, x_i^3, \dots, x_i^m) \quad (3.1)$$

$$y_i^{max} = \max(x_i^1, x_i^2, x_i^3, \dots, x_i^m) \quad (3.2)$$

$$d_i = |y_i^{max} - y_i^{min}| \quad (3.3)$$

$$z_i^{upper} = y_i^{max} + \alpha * d_i \quad (3.4)$$

$$z_i^{lower} = y_i^{min} - \alpha * d_i \quad (3.5)$$

$$z_i^{mid} = \text{mid}(y_i^{min}, y_i^{max}) \quad (3.6)$$

$$z_i^{avg} = \text{avg}(x_i^1, x_i^2, x_i^3, \dots, x_i^m) \quad (3.7)$$

Keterangan:

y_i^{min} = nilai minimum dari allele ke- i dari semua gen induk

y_i^{max} = nilai maximum dari allele ke- i dari semua gen induk

d_i = jarak antara y_i^{min} dan y_i^{max}

α = konstanta parameter BLX- α

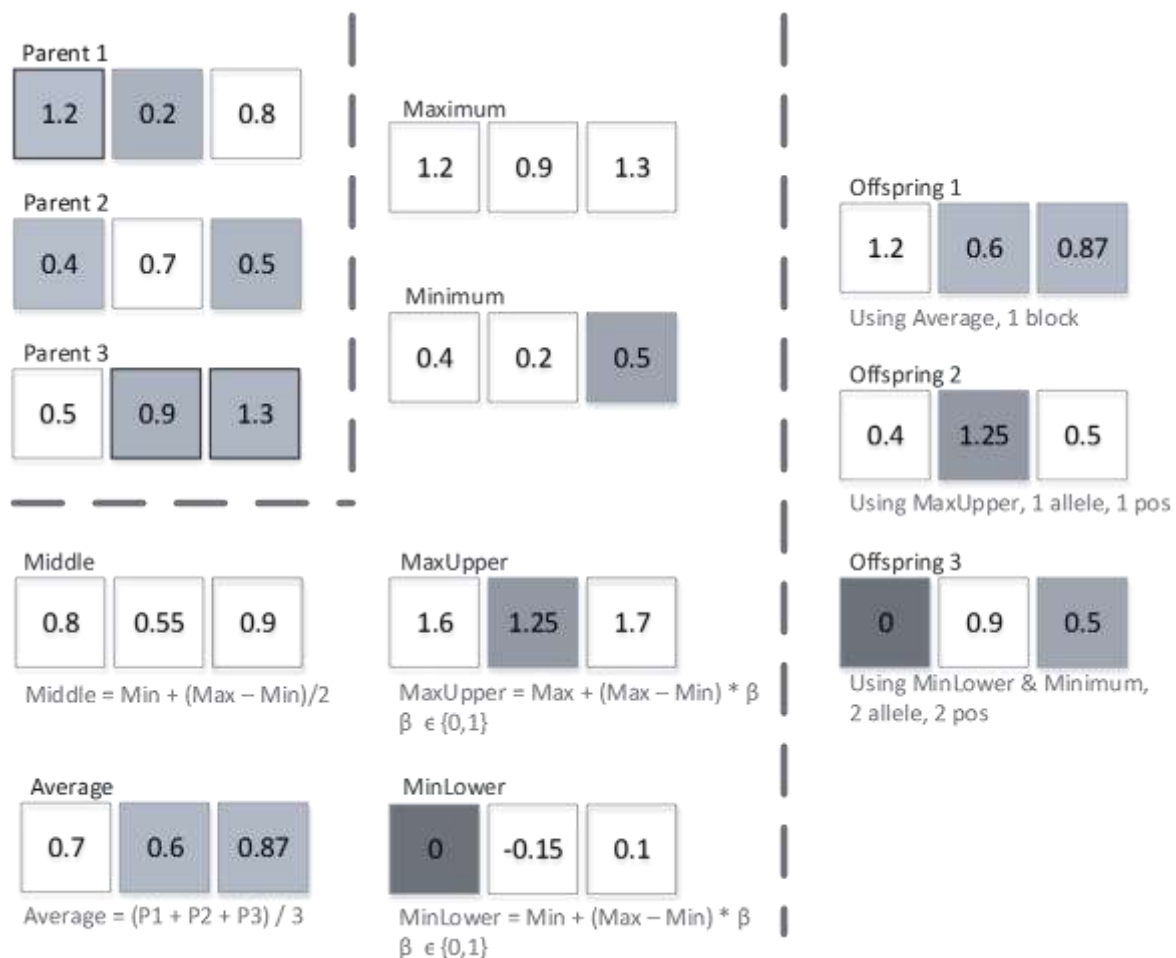
z_i^{upper} = nilai allele ke- i dari offspring upper Z^{upper}

z_i^{lower} = nilai allele ke- i dari offspring lower Z^{lower}

z_i^{mid} = nilai allele ke- i dari offspring middle Z^{middle}

$z_i^{average}$ = nilai allele ke- i dari offspring average $Z^{average}$

Untuk mendapat gambaran secara rinci, berikut ini ilustrasi perhitungan operasi rekombinasi BLX- α Multi-Induk di Gambar 3.6.



Gambar 3.6 Operasi Rekombinasi BLX- α Multi-Induk

3.2.2.3. Operasi Mutasi Aggregate Mate Pool

Operasi mutasi SGA hanya menggunakan nilai absolut dengan besaran konstan yang sama di semua iterasi generasi SGA. Penggunaan nilai absolut konstan ini mengakibatkan berkurangnya kemampuan adaptasi terhadap tingkat konvergensi SGA di setiap evolusi (iterasi). Penggunaan nilai konstan juga mengaburkan rentang jangkauan karena setiap permasalahan optimasi memiliki rentang ruang solusi yang berbeda-beda.

Operasi *Aggregate Mate Pool Mutation* (AMPM) bertujuan untuk memanfaatkan rentang nilai mutasi seluas-luasnya sesuai dengan rentang nilai yang tersedia. Mirip seperti metode operasi *Fine Mutation* (FM) yang dijelaskan dalam Bab 2.3.5 (Yoon & Kim, 2012), AMPM juga tidak menggunakan nilai absolut. Bila FM menggunakan nilai-nilai batas dari 2

induknya, AMPM menggunakan 5 gen referensi dalam *mate pool*. Teknik AMPM bertujuan untuk memperluas rentang nilai mutasi yang memungkinkan (*possible value range*). Hal tersebut memungkinkan untuk melakukan eksplorasi ruang solusi yang lebih luas lagi dan keluar dari kondisi *local optimum*.

Bila dipilih secara acak sebuah allele urutan ke- i , jumlah induk dalam *mate pool* adalah 5 dan gen induk yang terlibat adalah $X^1, X^2, X^3, \dots, X^5$, maka dapat dihitung nilai allele ke- i offspring z dengan persamaan sebagai berikut :

$$ym_i^{min} = \min(x_i^1, x_i^2, x_i^3, \dots, x_i^5) \quad (3.8)$$

$$ym_i^{max} = \max(x_i^1, x_i^2, x_i^3, \dots, x_i^5) \quad (3.9)$$

$$dm_i = |ym_i^{max} - ym_i^{min}| \quad (3.10)$$

Dari persamaan (3.8) s/d (3.10) maka persamaan (2.15) dapat disesuaikan menjadi :

$$z_i = z_i + Nd(0, \alpha_m * d_i) \quad (3.11)$$

Keterangan:

ym_i^{min} = nilai minimum dari allele ke- i dari semua gen dalam *mate pool*

ym_i^{max} = nilai maximum dari allele ke- i dari semua gen dalam *mate pool*

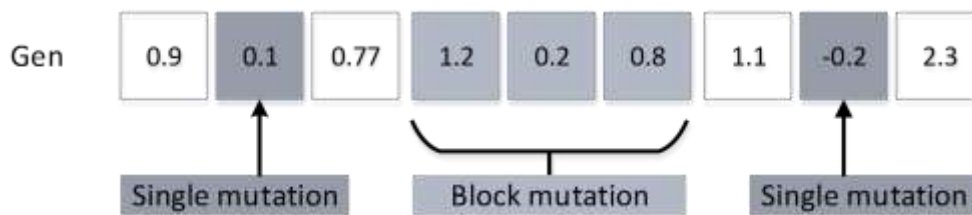
dm_i = jarak antara ym_i^{min} dan ym_i^{max}

α_m = konstanta parameter *Aggregate Mate Pool Mutation* dalam interval [0,1]

z_i = nilai allele ke- i dari offspring Z

Nd = fungsi normal distribution

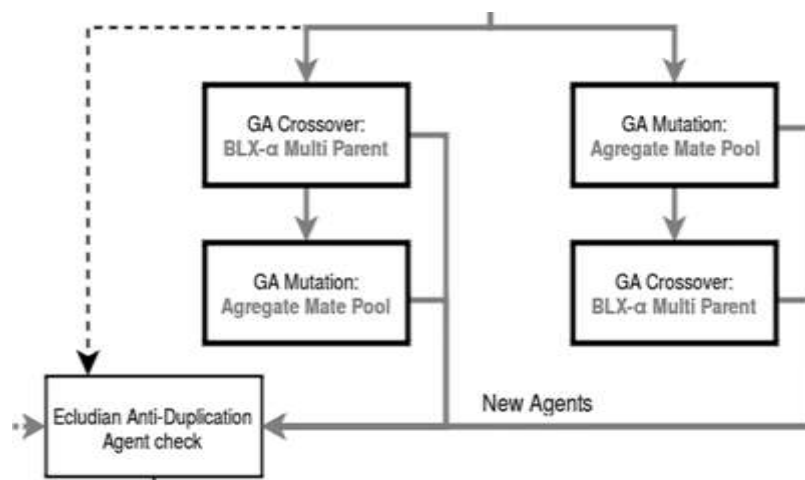
Sama seperti operasi rekombinasi, proses mutasi bisa dilakukan hanya pada satu allele (*single point mutation*), beberapa allele dalam posisi berbeda (*multi point mutation*), dalam satu blok (*block mutation*) atau kombinasi dari ketiganya (*combination mutation*). Untuk beberapa kasus, sangat dimungkinkan untuk melakukan operasi mutasi pada keseluruhan allele dalam gen bila jumlah allele sangat sedikit. Untuk lebih jelasnya dapat diamati ilustrasi berikut di Gambar 3.7.



Gambar 3.7 Operasi mutasi beberapa allele dan dalam blok

3.2.2.4. Operasi Rekombinasi-Mutasi Langsung

Dalam metode GA konvensional, operasi rekombinasi dan mutasi dilakukan dengan batasan tertentu, yaitu menggunakan konstanta peluang operasi rekombinasi dan peluang operasi mutasi. Penggunaan batasan peluang ini dapat mengakibatkan tidak efektifnya sebuah iterasi evolusi dalam GA, yaitu hanya terjadi operasi rekombinasi saja atau operasi mutasi saja. Sangat memungkinkan, sebuah proses mutasi dapat diikuti oleh sebuah proses rekombinasi saja. Untuk mengatasi hal ini, diusulkan sebuah teknik baru yang mengeksekusi operasi rekombinasi-mutasi dan mutasi-rekombinasi secara paralel. Teknik ini bertujuan untuk meningkatkan efektivitas setiap iterasi evolusi. Untuk lebih jelasnya dapat diamati Gambar 3.8.



Gambar 3.8 Alur operasi rekombinasi dan mutasi langsung

3.2.2.5. Rutin Anti Duplikasi Gen

Setiap operasi rekombinasi BLX- α Multi-Induk dan operasi mutasi Aggregate dapat menghasilkan beberapa turunan gen (*offspring*), maka perlu dipastikan terlebih dahulu bahwa semua offspring adalah offspring yang benar-benar berbeda dibandingkan dengan induknya dan gen lain dalam populasi. Pemeriksaan kemiripan antar gen menggunakan rumusan jarak

Euclidean seperti dijelaskan dalam Bab 2.3.1. Pemeriksaan kemiripan dilakukan terlebih dahulu kepada gen-gen induk dan gen-gen dalam populasi jodoh, baru kemudian dilakukan terhadap gen-gen lain dalam populasi.

Rutin anti duplikasi gen ini bertujuan untuk memastikan bahwa semua gen dalam populasi adalah gen yang berbeda, yaitu memiliki jarak Euclidean yang cukup jauh. Semakin besar jarak antar gen, maka akan semakin besar pula tingkat keragaman (*diversity*) sebuah populasi dan semakin lebar cakupan pencarian solusi yang dilakukan. Cakupan pencarian yang cukup lebar diperlukan untuk menghindari *local optimum* dan *premature convergence*.

3.2.2.6. Konstanta Normalisasi MinMax

Salah satu kelemahan MdAPE adalah nilainya menjadi sangat besar bila dataset berisi dengan angka yang sangat kecil atau mendekati nol. Sesuai dengan persamaan 2.9 dan 2.10, umumnya , nilai $S_{low} = 0$ dan $S_{high} = 1$, akan tetapi penggunaan nilai-nilai tersebut dapat mengakibatkan nilai MdAPE menjadi sangat besar karena pembagian dengan angka nol. Untuk itu diambil nilai $S_{low} = 0.4$ dan $S_{high} = 0.9$ untuk menghindari pembagian dengan nol.

3.2.2.7. Operasi Worst Replacement

Uraian secara detail dapat dijumpai pada Bab 2.3.7. Implementasi Operasi Worst Replacement (WR) dalam penelitian ini mengakibatkan gen terburuk dalam populasi selalu akan digantikan oleh gen-gen yang lebih baik, bila ditemukan gen *offspring* yang lebih baik hasil operasi rekombinasi dan mutasi. Dengan teknik ini, operasi seleksi Wide Tournament sangat memungkinkan untuk mendapatkan gen terburuk yang selalu berbeda di setiap generasi. Perubahan gen terburuk sangat penting artinya bagi operasi seleksi karena dengan perubahan tersebut dapat mengakibatkan berubahnya arah dan cakupan pencarian solusi yang berbeda di setiap generasi.

3.2.2.8. Perhitungan Fitness Function GA

Fitness Function yang digunakan di setiap gen GA menggunakan perumusan Median Absolute Percentage Error (MdAPE) yaitu kesalahan prediksi ANN seperti yang telah dijelaskan pada Bab 2.2.3.

3.2.2.9. Pemeriksaan Konvergensi

Sama seperti permasalahan optimasi pada umumnya, hibridasi ANN-WGA juga memerlukan sebuah perumusan *fitness function* sebagai kriteria kesesuaian dengan tujuan yang dijelaskan dalam Bab 2.4. Untuk itu, *fitness function* masing-masing gen dirumuskan sebagai kesalahan prediksi menggunakan rumus MdAPE pada Bab 2.2.3 diatas.

Pencarian konfigurasi optimum dilakukan dalam beberapa kali iterasi. Iterasi akan berhenti bila jumlah iterasi maksimum telah tercapai atau konvergensi telah terpenuhi. Bila dalam populasi terdapat m gen, konvergensi populasi dihitung dengan persamaan :

$$MdAPE_{avg} = avg(MdAPE^1, MdAPE^2, MdAPE^3, \dots MdAPE^m) \quad (3.12)$$

$$MdAPE_{min} = min(MdAPE^1, MdAPE^2, MdAPE^3, \dots MdAPE^m) \quad (3.13)$$

$$MdAPE_{max} = max(MdAPE^1, MdAPE^2, MdAPE^3, \dots MdAPE^m) \quad (3.14)$$

$$C_c = min(|MdAPE_{max} - MdAPE_{avg}|, |MdAPE_{min} - MdAPE_{avg}|) \quad (3.15)$$

Keterangan:

m = jumlah gen dalam populasi

$MdAPE_{max}$ = nilai *fitness function* terburuk milik gen terburuk

$MdAPE_{min}$ = nilai *fitness function* terbaik

$MdAPE_{avg}$ = rata-rata *fitness function* seluruh gen dalam populasi

C_c = kriteria konvergensi

Iterasi evolusi WGA akan berhenti bila nilai $C_c < 0.000001$, yang berarti bahwa semua gen dalam populasi memiliki *fitness function* yang hampir sama.

3.2.3. Prediksi Liquefaction menggunakan ANN-WGA

Setelah melakukan pelatihan ANN menggunakan WGA dan mendapatkan konfigurasi bobot koneksi ANN yang paling baik, proses selanjutnya dilanjutkan dengan melakukan regresi menggunakan data liquefaction yang baru. Proses regresi menggunakan teknik Feedforward yang telah dijelaskan dalam Bab 2.2 dan Bab 2.4. Keluaran dari proses regresi adalah besaran nilai kedalaman maksimum Liquefaction.

3.3. Pembuatan Perangkat Lunak

Sebelum melakukan pengujian empiris, sebuah perangkat lunak dibuat terlebih dahulu sesuai dengan spesifikasi model di Bab 3.2. Pembuatan perangkat lunak akan dilaksanakan dalam lingkungan pengembangan dengan spesifikasi pada Tabel 3.2.

Tabel 3.2 Spesifikasi lingkungan pengembangan

Parameter	Spesifikasi
Sistem Operasi	Linux Debian 8.5 64bit
RAM	8 GB
Processor	Intel i5-4690
Bahasa Pemrograman	C++
Library	Open MPI, PCG Random Generator
IDE	Netbeans 8.2

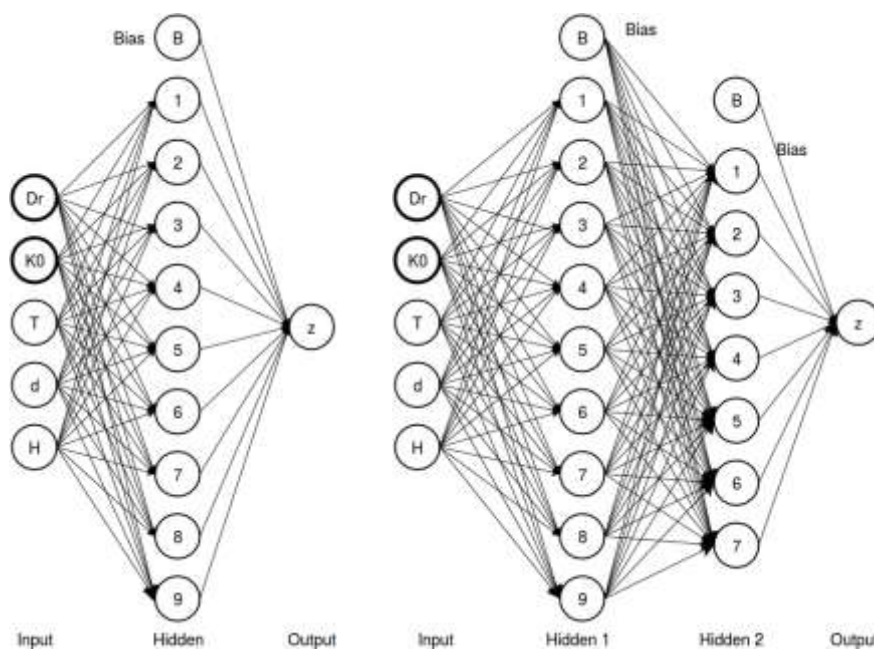
3.4. Pengujian Sistem (Empiris)

Data yang digunakan dalam penelitian ini bersumber dari beberapa paper jurnal serta laporan tesis dari penelitian terdahulu. Setiap data dari paper yang berbeda dianggap sebagai satu kasus pengujian tersendiri karena masing-masing mewakili kondisi perairan yang berbeda. Sampai saat ini terdapat 18 paper yang dapat digunakan sebagai rujukan untuk mendapatkan data dasar. Sebagian besar paper-paper tersebut memiliki data yang mirip sama, sehingga diputuskan untuk menggunakan 6 paper dengan karakteristik data yang benar-benar berbeda, terutama pada data karakteristik tanah. Tabel 3.3 adalah daftar dataset yang digunakan dalam Uji Coba.

Tabel 3.3 Dataset untuk Uji Coba

No	Dataset	Jumlah Atribut	Output	Jumlah Data
1	Liquefaction Dataset 1	6	1	1125
2	Liquefaction Dataset 2	6	1	1350
3	Liquefaction Dataset 3	6	1	2520
4	Liquefaction Dataset 4	6	1	1200
5	Liquefaction Dataset 5	6	1	1800
6	Liquefaction Dataset 6	6	1	2100
7	Liquefaction Dataset 7	6	1	1152

Skenario pengujian liquefaction adalah kombinasi dari 7 dataset, 3 metode pelatihan ANN dan 2 jenis arsitektur ANN. Masing-masing dataset memiliki parameter kekuatan tanah dan beban gelombang yang berbeda. Metode pelatihan yang digunakan adalah BP, GA dan WGA. Arsitektur ANN yang akan digunakan adalah satu hidden layer ($5_{(i)}-9_{(h)}-1_{(o)}$), 5 input, 9 hidden node, 1 output) dan 2 hidden layer ($5_{(i)}-9_{(h)}-7_{(h)}-1_{(o)}$), 5 node input, 9 hidden node layer 1, 7 hidden node layer 2, 1 output node). Konfigurasi setiap hidden layer secara otomatis ditambahkan juga 1 (satu) bias node. Untuk lebih jelasnya, bisa diamati Gambar 3.9.

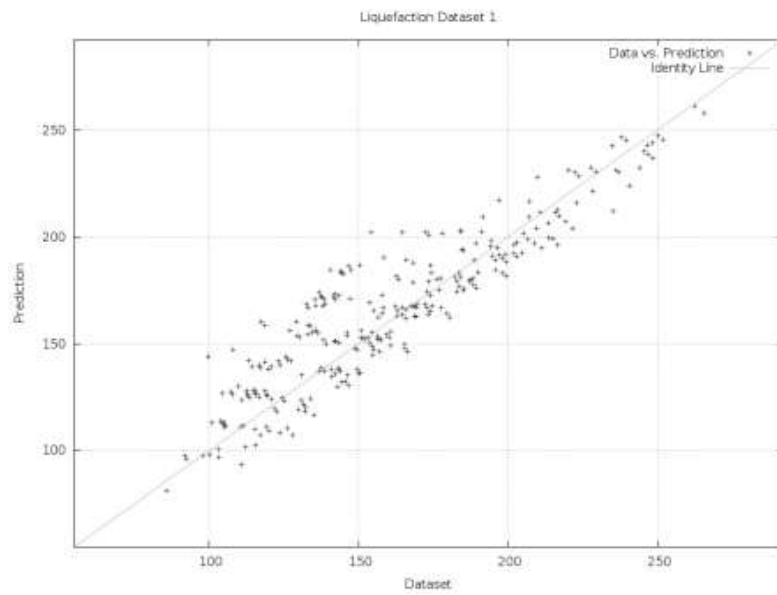


Gambar 3.9 Konfigurasi ANN untuk prediksi Liquefaction,
 $5_{(i)}-9_{(h)}-1_{(o)}$ di sebelah kiri dan $5_{(i)}-9_{(h)}-7_{(h)}-1_{(o)}$ di sebelah kanan.

3.5. Analisa Hasil Pengujian

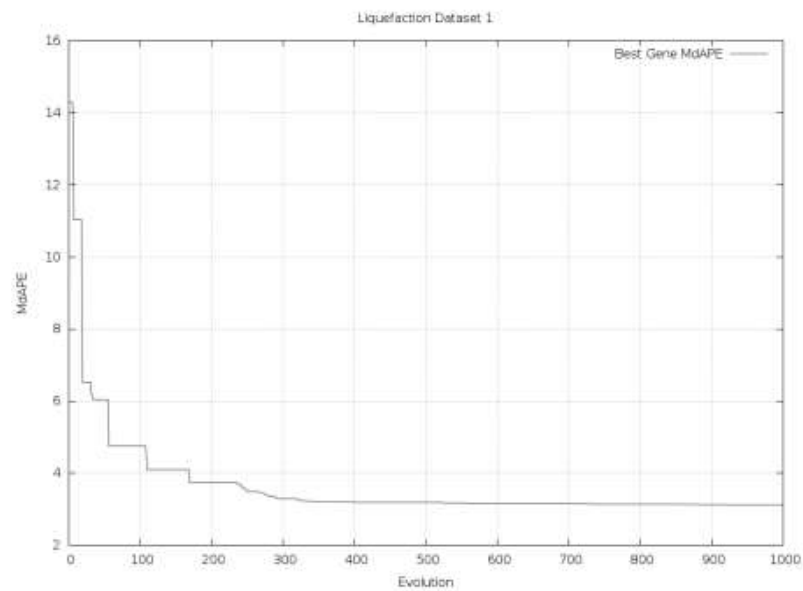
Untuk mengetahui kinerja metode optimasi WGA dalam pelatihan ANN, digunakan rumus MdAPE yang telah dijelaskan dalam Bab 2.2.3 diatas. Semakin kecil nilai MdAPE menunjukkan kinerja WGA yang semakin baik. Selain menggunakan MdAPE, kinerja WGA dapat diketahui menggunakan diagram scattered dan diagram konvergensi.

Diagram scattered adalah gambaran titik-titik sebaran antara nilai pada dataset di sumbu x dan hasil prediksi di sumbu y . Diagram scattered menggunakan garis diagonal, yang disebut sebagai garis *identity* untuk melihat penyimpangan antara nilai dataset dan hasil prediksi. Contoh diagram scattered dapat dilihat dalam Gambar 3.10.



Gambar 3.10 Contoh Diagram Scattered

Diagram konvergensi adalah gambaran MdAPE gen terbaik di setiap evolusi. Sumbu x pada diagram konvergensi adalah iterasi evolusi dan sumbu y adalah MdAPE gen terbaik pada iterasi tertentu. Gambar 3.11 adalah contoh diagram konvergensi.



Gambar 3.11 Contoh Diagram Konvergensi

[Halaman ini sengaja dikosongkan]

BAB 4

UJICOBAN DAN PEMBAHASAN

Pada Bab 4 ini akan dijelaskan mengenai lingkungan uji coba dan hasil-hasil yang didapatkan setelah melakukan uji coba.

4.1. Lingkungan Uji Coba

Seperti telah dijelaskan dalam Bab 3.3 diatas, sebuah aplikasi dikembangkan secara khusus untuk mengetahui kinerja WGA dalam pelatihan ANN. Kemudian, aplikasi tersebut digunakan dalam pengujian empiris. Keluaran dari aplikasi tersebut digunakan untuk membandingkan kinerja WGA dibandingkan dengan metode pelatihan ANN yang lain, yaitu BP dan SGA. Tabel 4.1 adalah spesifikasi lingkungan pengujian empiris.

Tabel 4.1 Spesifikasi Lingkungan Pengujian

Komponen	Spesifikasi
Processor	Intel i7-4720HQ
RAM	8 GB
HDD	1 TB
Sistem Operasi	Linux Debian 8.6 Jessie 64bit
Kernel	4.7.0-0.bpo.1-amd64
Shell	Bash 4.3.30

4.2. Spesifikasi Dataset

Telah dijelaskan dalam Bab 3.2.1 bahwa dataset yang digunakan dalam pengujian adalah dataset sintetis yang dibangkitkan berdasarkan data lingkungan dari penelitian terdahulu. Tabel 4.2 adalah spesifikasi dataset yang digunakan dalam pengujian.

Tabel 4.2 Spesifikasi Dataset

Dataset	Referensi	Atribut	Nilai Awal	Variasi Nilai
1	(Nataraja & Gill, 1983)	T	15 detik	12, 13, 14, 15, 16
		d	70 m	10, 20, 30, 50, 70
		H	25 m	5, 10, 15, 20, 25
		K0	0.18	0.16, 0.18, 0.20
		Dr	0.35-0.65	0.65, 0.75, 0.85

2	(Liao, et al., 2015)	T	12 detik	8, 10, 12
		d	30 m	5, 10, 15, 20, 25, 30
		H	2.5 m	0.5, 1.0, 1.5, 2.0, 2.5
		K0	0.18	0.15, 0.18, 0.21
		Dr	0.35-0.65	0.35, 0.45, 0.55, 0.65, 0.75
3	(Liao, et al., 2015)	T	5 detik	3, 4, 5, 6
		d	5 m	2, 3, 4, 5, 6
		H	2.5 m	1.8, 2.0, 2.4, 2.6, 2.8, 3.0
		K0	0.18	0.15, 0.18, 0.21
		Dr	0.35-0.65	0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95
4	(Cha, et al., 2009)	T	8 detik	5, 6, 7, 8, 9
		d	50 m	30, 40, 50, 60
		H	7.5 m	5.5, 6.0, 6.5, 7.0, 7.5
		K0	0.18	0.15, 0.18, 0.21
		Dr	0.35-0.65	0.35, 0.45, 0.55, 0.65
5	(Cha, et al., 2006)	T	8-20 detik	8, 10, 12, 16, 20
		d	20-80 m	20, 35, 50, 65, 80
		H	0.8-10 m	3, 4, 5, 6, 7, 8
		K0	0.18	0.15, 0.18, 0.21
		Dr	0.35-0.65	0.35, 0.45, 0.55, 0.65
6	(Liao, et al., 2015)	T	5 detik	4, 4.5, 5, 5.5
		d	5 m	2, 3, 4, 5, 6
		H	2.5 m	2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6
		K0	0.18	0.15, 0.18, 0.21
		Dr	0.35-0.65	0.70, 0.75, 0.80, 0.85, 0.90
7	(Jeng, 1997)	T	15 detik	12, 13, 14, 15
		d	70 m	55, 60, 65, 70
		H	24 m	16, 18, 20, 22, 24, 26
		K0	0.18	0.15, 0.18, 0.21
		Dr	0.35-0.65	0.35, 0.45, 0.55, 0.65

Dari setiap data di kolom variasi nilai di Tabel 4.2 kemudian dibangkitkan keseluruhan dataset dalam proses iterasi sebanyak jumlah data dalam variasi nilai.

4.3. Skenario Uji Coba Prediksi Liquefaction

Pada Bab 3.4 dijelaskan bahwa pengujian akan dilakukan dalam beberapa skenario, setiap skenario adalah kombinasi dari 7 dataset, 3 jenis metode pelatihan ANN dan 2 jenis arsitektur ANN. Untuk memudahkan pengujian dan pembahasan, disusun 2 skenario besar berdasarkan jenis arsitektur ANN antara lain :

- Skenario Uji Coba 1:
Uji coba dilakukan pada ANN dengan 1 hidden layer, bertujuan untuk mengetahui kinerja WGA dalam proses pelatihan ANN yang berkonfigurasi sederhana.
- Skenario Uji Coba 2:
Uji coba dilakukan pada ANN dengan 2 hidden layer, bertujuan untuk mengetahui kinerja WGA dengan konfigurasi ANN yang jauh lebih kompleks.

Dari sudut pandang bidang optimasi, kedua skenario memiliki tingkat kompleksitas yang sangat berbeda. Skenario 1 jauh lebih sederhana karena panjang gen dalam skenario 1 memiliki 54 nilai allele dalam setiap gen. Sementara skenario 2 memiliki panjang gen sampai 115 nilai allele, dua kali lipat bila dibandingkan dengan skenario 1. Secara umum dapat dikatakan kompleksitas pengujian skenario 2 adalah dua kali lebih kompleks bila dibandingkan dengan skenario 1. Tabel 4.3 s/d Tabel 4.6 adalah spesifikasi dan besaran konstanta yang digunakan dalam pengujian, baik untuk Skenario 1 dan Skenario 2.

Tabel 4.3 Paramater dan Konstanta Umum

Parameter dan Konstanta	Nilai	Keterangan
Jumlah eksperimen independen	5	Jumlah eksperimen yang dilakukan untuk setiap metode pelatihan
Iterasi maksimum	1000	Jumlah iterasi setiap eksperimen independen
Metode normalisasi data	MaxMin	Dijelaskan dalam Bab 2.2.6 dan 3.2.2.6, nilai $S_{low} = 0.4$ dan $S_{high} = 0.9$
Proporsi data latih	75%	
Proporsi data uji coba	25%	
Kriteria konvergensi	0.000001	Dijelaskan dalam Bab 3.2.2.9

Parameter dan konstanta yang terdapat dalam Tabel 4.3 digunakan oleh ketiga metode pelatihan, yaitu BP, GA dan WGA.

Tabel 4.4 Parameter dan Konstanta BP

Parameter dan Konstanta	Nilai
Learning Rate	0.01
Momentum	0.5
K-Folds	7

Tabel 4.5 Parameter dan Konstanta GA

Parameter dan Konstanta	Nilai
Jumlah Populasi	20
Peluang Rekombinasi	0.5
Peluang Mutasi	0.2
Operator Seleksi	Binary Tournament
Operator Rekombinasi	BLX- α
Operator Mutasi	Fine Mutation

Tabel 4.6 Parameter dan Konstanta WGA

Parameter dan Konstanta	Nilai
Jumlah Populasi	20
Operator Seleksi	Wide Tournament
Operator Rekombinasi	Multi Parent BLX- α
Operator Mutasi	Aggregate Mate Pool Mutation
Operator Penggantian	Worst Replacement
Uji Keragaman Populasi	Euclidean Distance

4.4. Hasil Uji Coba Skenario 1

Seperti telah dijelaskan dalam Bab 4.2 dan 3.2.1, pengujian akan menggunakan 7 dataset (Dataset 1 s/d 7) dan 3 metode pelatihan ANN (BP, GA, WGA). Pada Tabel 4.7 adalah ringkasan hasil uji coba Skenario 1 pada dataset 1-7.

Tabel 4.7 Ringkasan Hasil Pengujian Skenario 1

Dataset	Jumlah Data			ANN Arch	MdAPE		
	Total	Train	Test		ANN-BP	ANN-GA	ANN-WGA
Liquefaction Dataset 1	1,125	844	281	5:9:1	5.052	4.065	3.189
Liquefaction Dataset 2	1,350	1,013	337	5:9:1	6.544	4.358	3.241
Liquefaction Dataset 3	2,520	1,890	630	5:9:1	3.032	2.620	1.410
Liquefaction Dataset 4	1,200	900	300	5:9:1	6.881	5.082	4.346
Liquefaction Dataset 5	1,800	1,350	450	5:9:1	4.670	3.526	3.050
Liquefaction Dataset 6	2,100	1,575	525	5:9:1	6.257	3.067	2.454
Liquefaction Dataset 7	1,152	864	288	5:9:1	5.072	3.760	3.176
Average MdAPE of All Datasets					5.358	3.783	2.981

Keterangan:

Total = jumlah keseluruhan data

Train = jumlah data pelatihan ANN

Test = jumlah data uji ANN

ANN Arch = konfigurasi jumlah node dalam setiap layer ANN

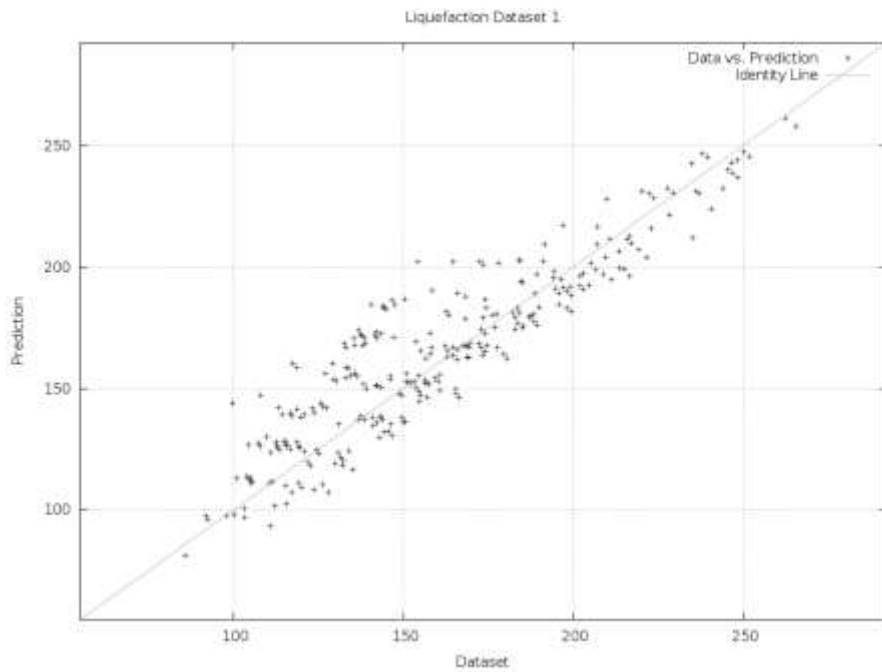
5:9:1 = 5 node input, 9 hidden node + 1 bias node, 1 output node

MdAPE = Median Absolute Percentage Error (Bab 2.2.3) dalam persen (%)

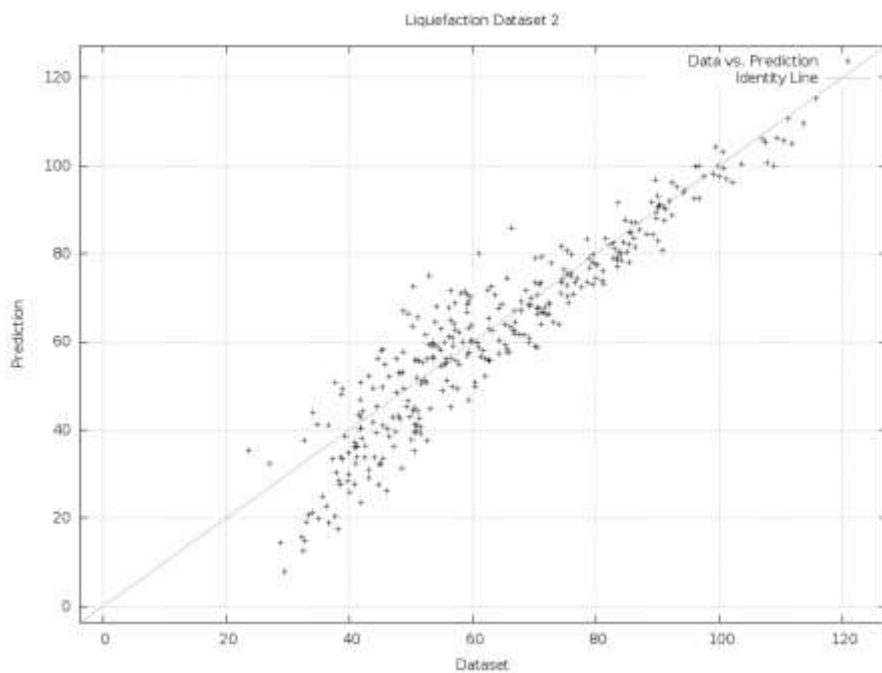
Dari hasil uji coba pada Tabel 4.7 dapat disimpulkan bahwa ANN-WGA dapat menghasilkan MdAPE terkecil 1.410% pada Liquefaction Dataset 3 dan MdAPE terbesar 4.346% pada Dataset 4. Secara umum untuk semua dataset, ANN-WGA memberikan kinerja lebih baik dibandingkan ANN-BP dan ANN-GA yaitu dengan rata-rata MdAPE sebesar 2.981%.

4.4.1. Digram Scattered Uji Coba Skenario 1

Gambar 4.1 s/d Gambar 4.7 adalah diagram scattered dari hasil setiap proses uji coba menggunakan ANN-WGA. Nilai Dataset dan Prediction dalam scattered diagram dalam satuan meter. Dari hasil uji coba dataset 1 pada Gambar 4.1 menunjukkan penyebaran titik-titik z cukup merata pada semua rentang kecuali pada $z > 200m$ yang lebih renggang. Secara umum jarak semua titik ke garis idenity tidak terlalu jauh, sehingga menghasilkan MdAPE 3.189%.

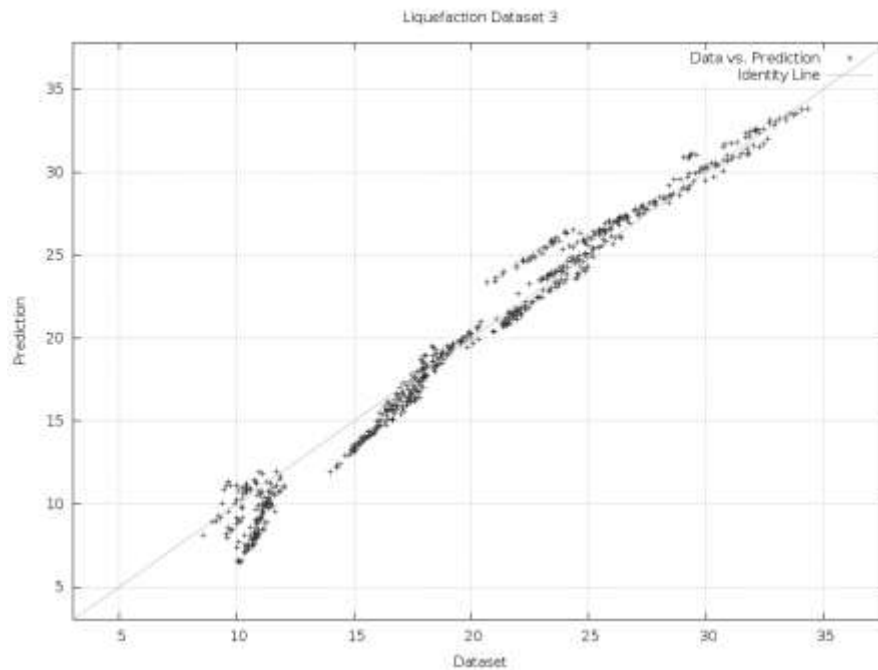


Gambar 4.1 Diagram scattered pada skenario 1 dataset 1



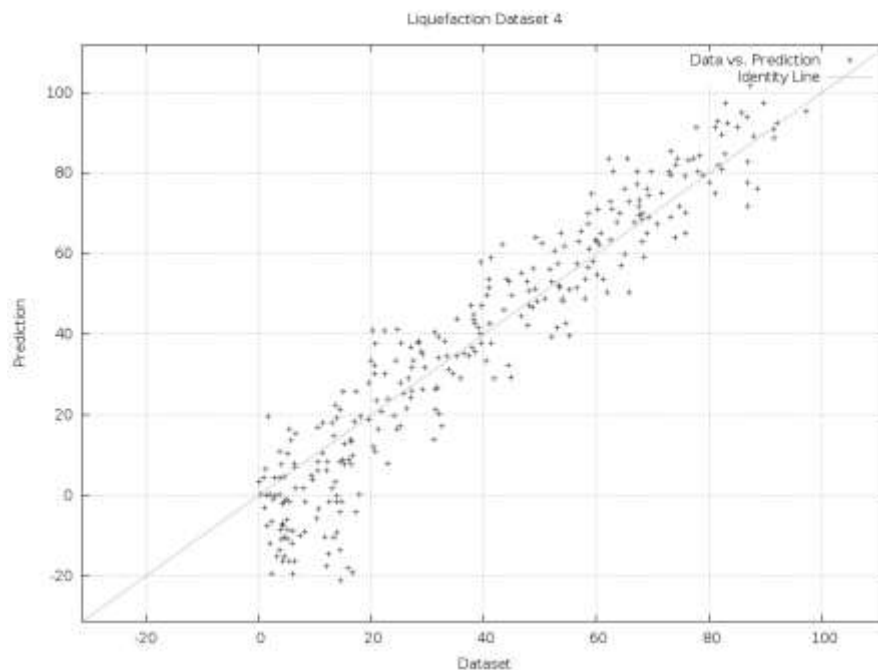
Gambar 4.2 Diagram scattered pada skenario 1 dataset 2

Nilai MdAPE untuk dataset 2 (3.241%) sedikit lebih besar dibandingkan dataset 1 (3.189%), hal ini disebabkan oleh penyebaran titik-titik yang cukup jauh dari garis identity terutama pada $z < 50m$. Pada $z > 50m$ penyebaran titik-titik mulai mendekati garis identity (Gambar 4.2).



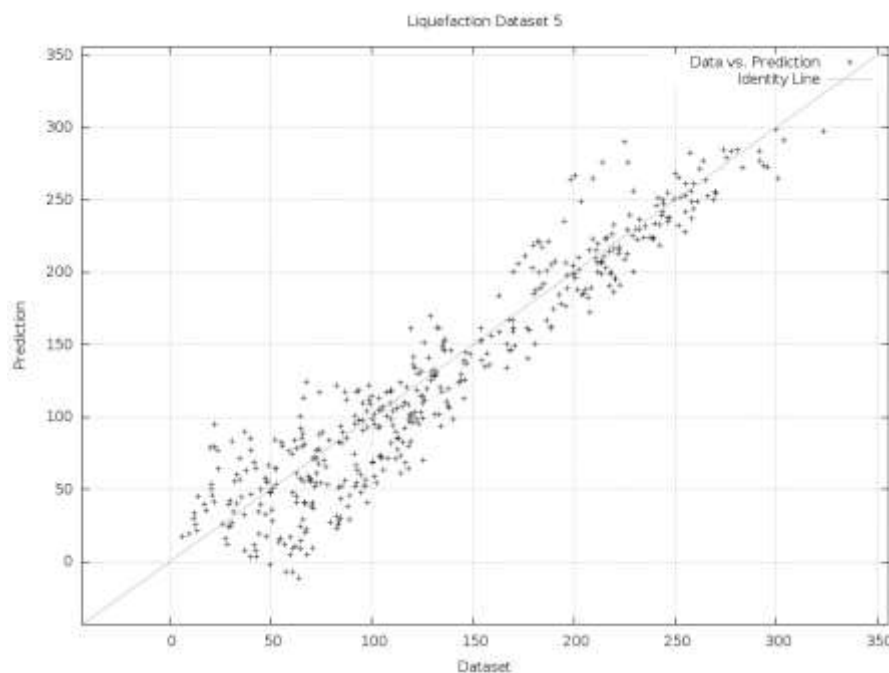
Gambar 4.3 Diagram scattered pada skenario 1 dataset 3

Uji coba pada dataset 3 menghasilkan nilai MdAPE yang terkecil dibandingkan pada dataset lainnya, yaitu 1.410%. Pada Gambar 4.3 penyebaran titik-titik rapat dan dekat garis identity bahkan sangat dekat dibandingkan dengan diagram lainnya. Walaupun ada kekosongan data z pada rentang $12 < z < 14$ akan tetapi ANN-WGA memberikan MdAPE terbaik.



Gambar 4.4 Diagram scattered pada skenario 1 dataset 4

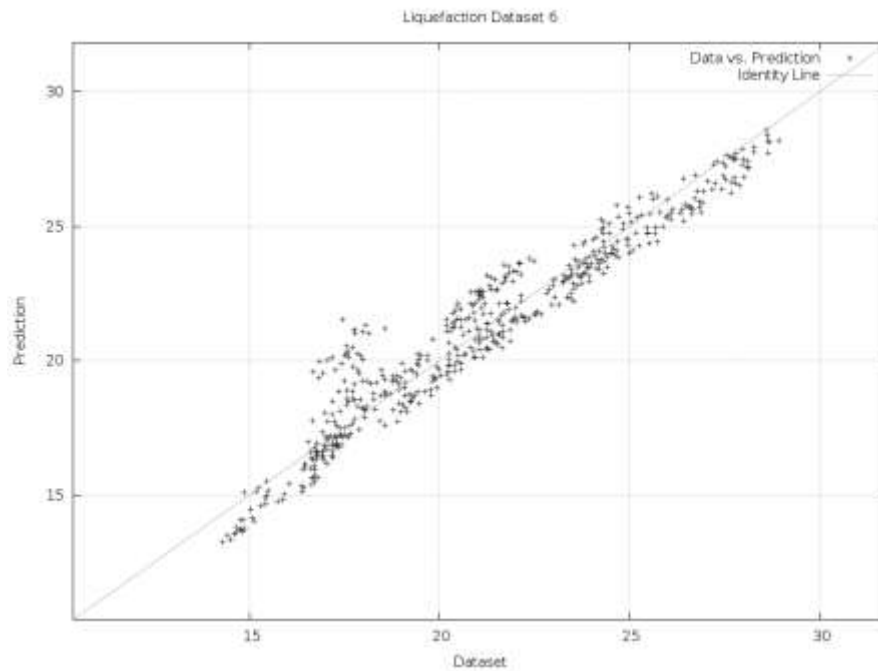
Nampak pada Gambar 4.4, beberapa titik di rentang nilai z antara 0-20m memiliki simpangan yang cukup lebar. Bila dibandingkan dengan diagram scattered yang lain, yaitu Gambar 4.1 s/d Gambar 4.7, hasil uji coba skenario 1 pada dataset 4 ini menunjukkan sebaran data yang lebih baik. Data prediksi tersebar merata mulai nilai $z = 0m$ s/d $z = 100m$. Salah satu kelemahan uji coba pada dataset 4 adalah jumlah data yang digunakan sangat sedikit yaitu 1200 baris data, sehingga pada uji coba skenario 1 ini, hasil dataset 4 memiliki error MdAPE relatif tinggi (4.346%) dibandingkan dataset lain.



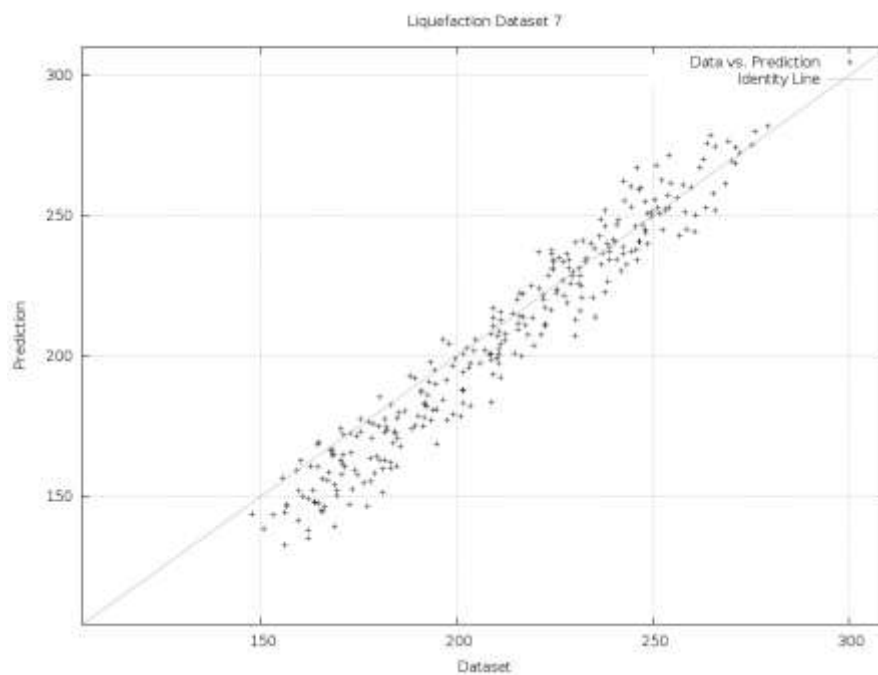
Gambar 4.5 Diagram scattered pada skenario 1 dataset 5

Sama seperti uji coba skenario 1 pada dataset 3 (Gambar 4.3), pada uji coba skenario 1 dataset 5 (Gambar 4.5) juga terdapat sedikit kekosongan data z pada rentang $150 < z < 170$. Secara umum, penyebaran titik-titik pada dataset 5 cukup merata, kecuali pada rentang $z < 100m$.

Pada uji coba skenario 1 untuk dataset 6 (Gambar 4.6), penyebaran titik-titik Prediction vs. Dataset cukup merata di seluruh rentang nilai z dan berjarak cukup dekat dengan garis identity. Bahkan, beberapa titik cukup dekat jaraknya dengan titik lainnya sehingga nampak ada penebalan di beberapa lokasi. Kondisi inilah yang menyebabkan nilai MdAPE yang dihasilkan cukup kecil, yaitu 2.454%. MdAPE dataset 6 adalah MdAPE terkecil kedua setelah MdAPE dataset 3.



Gambar 4.6 Diagram scattered pada skenario 1 dataset 6



Gambar 4.7 Diagram scattered pada skenario 1 dataset 7

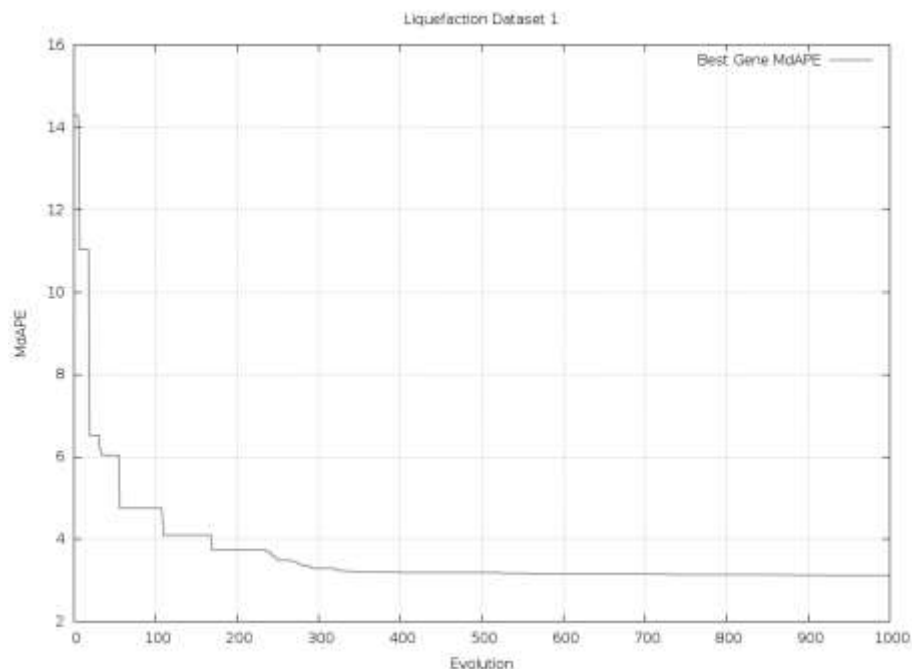
Uji coba skenario 1 pada dataset 7 dilakukan dengan data yang relatif sedikit, yaitu hanya 1152 baris data (Tabel 4.7). Walaupun demikian diagram scattered pada Gambar 4.7

menunjukkan hasil yang relatif baik dengan MdAPE sebesar 3.176%. Penyebaran titik-titik nilai Prediction vs. Dataset cukup rapat dengan garis identity dan cukup merata pada semua rentang nilai z , baik pada z nilai kecil maupun besar.

Dari scattered diagram pada Gambar 4.1 s/d Gambar 4.7, secara umum dapat disimpulkan bahwa hasil setiap prediksi ANN-WGA sudah mendekati garis identity, yang berarti bahwa kesalahan (error) prediksi ANN-WGA sangat kecil. Penyebaran titik-titik Prediction vs. Dataset juga cukup merata pada semua rentang nilai z . Hal ini menunjukkan bahwa dataset yang digunakan dalam uji coba skenario 1 sudah cukup baik.

4.4.2. Diagram Konvergensi Uji Coba Skenario 1

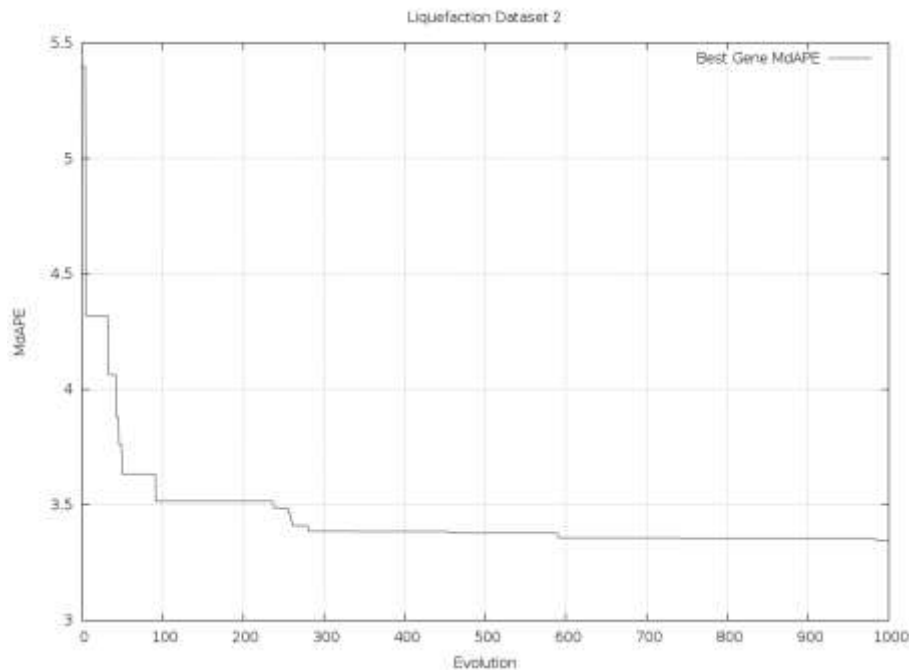
Uji coba skenario 1 pada dataset 1-7 memberikan beberapa hasil yang dapat dianalisa lebih jauh. Hasil pertama adalah diagram scattered yang menunjukkan penyebaran hasil prediksi dibandingkan dengan nilai dataset dan garis identity. Dari diagram scattered dapat diamati pola penyebaran nilai prediksi setiap data dari keseluruhan dataset. Hasil kedua uji coba berupa diagram konvergensi (Gambar 4.8 s/d Gambar 4.14). Diagram konvergensi berfungsi untuk mengetahui laju tingkat konvergensi (*convergence level*) di setiap iterasi evolusi ANN-WGA.



Gambar 4.8 Diagram konvergensi pada skenario 1 dataset 1

Pada Gambar 4.8, konvergensi mulai dicapai pada evolusi ke-300 dan pada evolusi ke-400 nilai MdAPE tidak lagi berkurang. Hal ini berarti bahwa ANN-WGA telah mencapai

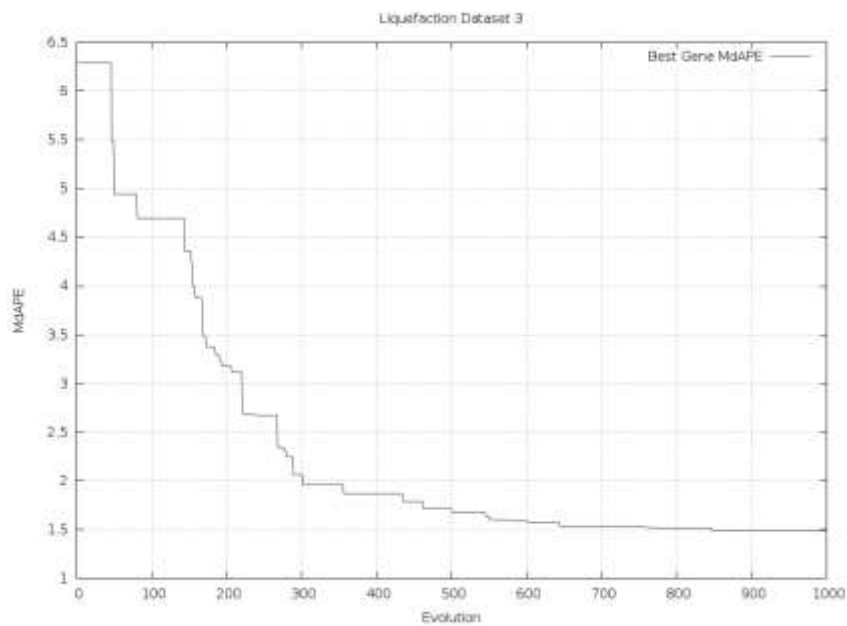
kondisi konvergensi pada iterasi ke-400. Penurunan MdAPE secara drastis pada evolusi < 200 terjadi karena operasi-operasi pada ANN-WGA terutama operasi *Aggregate Mate Pool Mutation* (dijelaskan dalam BAB 3.2.2.3) bisa melompat cukup jauh untuk menghasilkan offspring yang lebih baik.



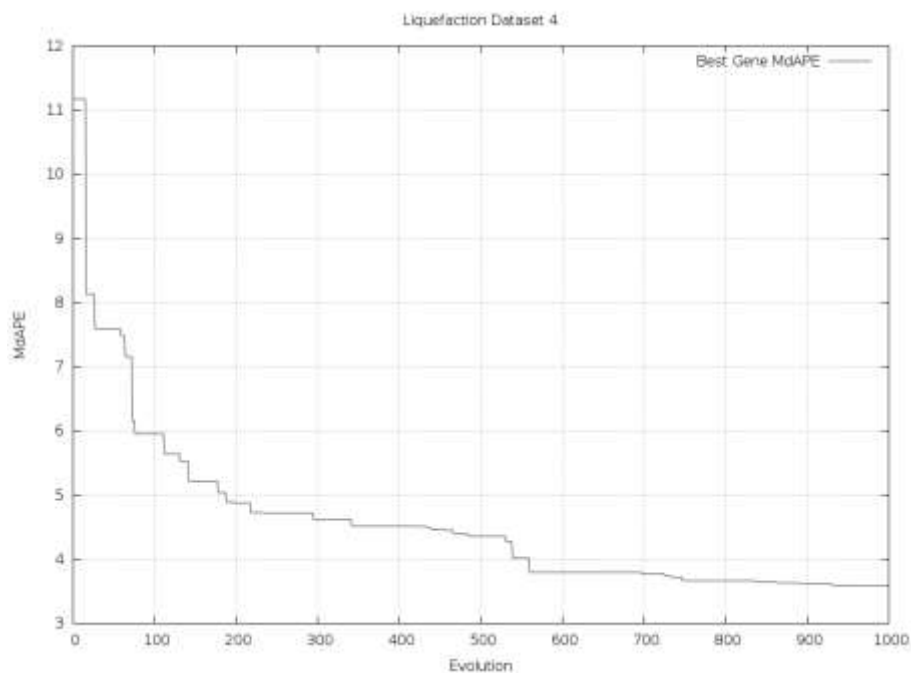
Gambar 4.9 Diagram konvergensi pada skenario 1 dataset 2

Pada uji coba skenario 1 pada dataset 2 (Gambar 4.9), konvergensi mulai terjadi pada evolusi ke-300. MdAPE tidak lagi mengalami penurunan pada evolusi ke-600. Hal ini bisa disimpulkan bahwa konvergensi ANN-WGA terjadi pada evolusi ke-600. Walaupun terjadi penurunan pada evolusi ke-990, akan tetapi nilainya sangat kecil sehingga bisa diabaikan.

Berbeda dengan hasil uji coba dataset sebelumnya, konvergensi uji coba skenario 1 pada dataset 3 menunjukkan tren yang lebih landai (Gambar 4.10). MdAPE ANN-WGA turun secara gradual dan pelan di setiap evolusi, hampir tidak ada lompatan penurunan MdAPE. Dataset 3 adalah dataset berukuran besar, berisi 2520 baris data (Tabel 4.7) sehingga penurunan MdAPE kesulitan untuk menurunkan MdAPE secara drastis. Namun demikian, uji coba skenario 1 pada dataset 3 ini memberikan hasil terbaik dengan MdAPE sebesar 1.410%. Konvergensi mulai tercapai pada evolusi ke-700 dan pada evolusi ke-900 sudah tidak terjadi lagi penurunan MdAPE sehingga bisa disimpulkan bahwa untuk dataset 3, konvergensi tercapai pada evolusi ke-900.

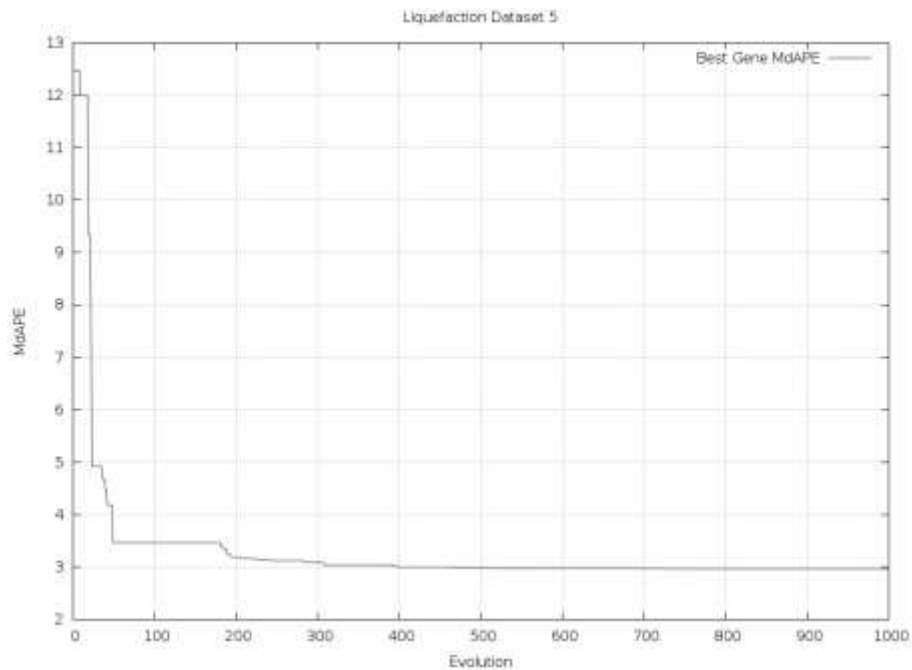


Gambar 4.10 Diagram konvergensi pada skenario 1 dataset 3



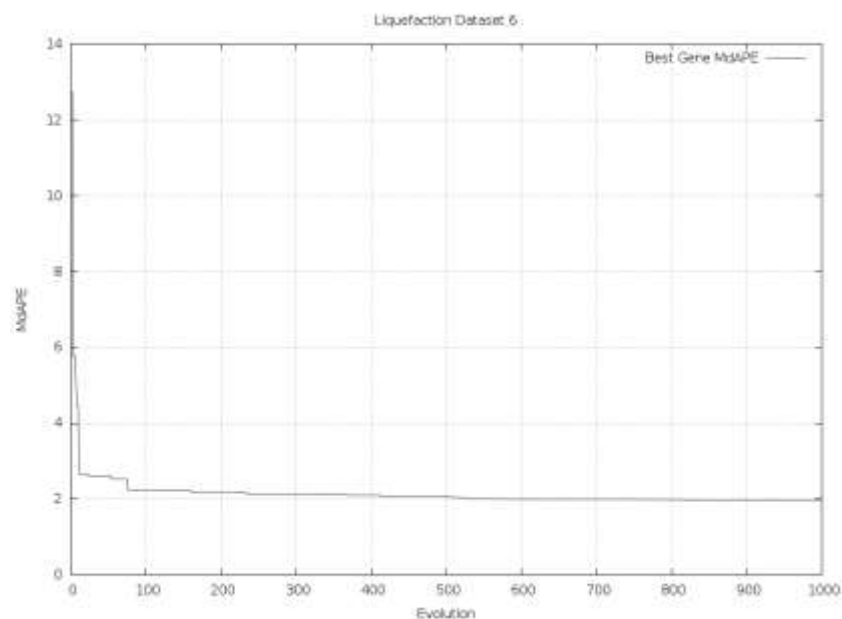
Gambar 4.11 Diagram konvergensi pada skenario 1 dataset 4

Mirip dengan uji coba terdahulu pada uji coba skenario 1 dataset 3 (Gambar 4.10), konvergensi pada dataset 4 juga terjadi secara gradual dan perlahan (Gambar 4.11). Penurunan MdAPE terjadi sangat drastis pada awal evolusi yaitu pada evolusi < 100 . Konvergensi mulai terjadi pada evolusi ke-600 dan pada evolusi ke-800 sudah tidak lagi terjadi penurunan MdAPE yang cukup signifikan. Dapat disimpulkan bahwa konvergensi terjadi pada evolusi ke-800.



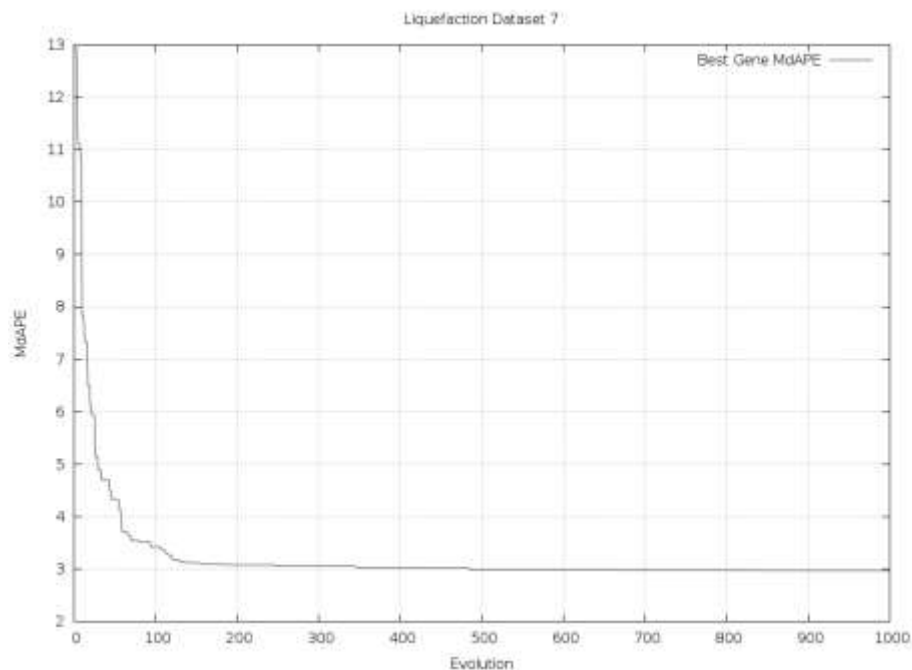
Gambar 4.12 Diagram konvergensi pada skenario 1 dataset 5

Uji coba skenario 1 pada dataset 5 (Gambar 4.12) menunjukkan lonjakan penurunan MdAPE yang sangat besar, terutama pada evolusi < 50 , kemudian secara bertahap mencapai kondisi konvergensi pada evolusi ke-200. Pada evolusi ke-400 sudah tidak ada lagi penurunan MdAPE yang cukup signifikan, sehingga bisa disimpulkan bahwa konvergensi ANN-WGA terjadi pada evolusi ke-400.



Gambar 4.13 Diagram konvergensi pada skenario 1 dataset 6

Mirip seperti uji coba pada dataset 5 (Gambar 4.12), uji coba skenario 1 pada dataset 6 juga mengalami penurunan MdAPE secara drastis di awal evolusi, terutama di evolusi < 30 (Gambar 4.13). Konvergensi mulai terjadi pada evolusi ke-100 dan pada evolusi ke-600 tidak lagi ada penurunan MdAPE yang cukup signifikan. Dapat disimpulkan ANN-WGA mengalami konvergen pada evolusi ke-600.



Gambar 4.14 Diagram konvergensi pada skenario 1 dataset 7

Uji coba skenario 1 dataset 7 menunjukkan hasil yang mirip sama dengan 2 uji coba sebelumnya (Gambar 4.12 dan Gambar 4.13). Penurunan MdAPE terjadi pada awal evolusi, yaitu evolusi < 30 , kemudian secara bertahap turun sampai evolusi ke-200 (Gambar 4.14). Pada evolusi ke-500 sudah tidak terjadi penurunan MdAPE yang cukup signifikan, sehingga dapat disimpulkan bahwa konvergensi terjadi pada evolusi ke-500.

Diagram konvergensi dari Gambar 4.8 s/d Gambar 4.14 menunjukkan bahwa sebagian besar uji coba mengalami penurunan MdAPE secara drastis pada awal evolusi, kecuali untuk dataset 3 (Gambar 4.10) dan dataset 4 (Gambar 4.11). Seluruh dataset dapat mencapai konvergensi kurang dari jumlah evolusi maksimum, yaitu 1000 iterasi evolusi (Tabel 4.3). Dari hasil-hasil tersebut dapat disimpulkan bahwa seluruh operasi dalam WGA terutama operasi Anti-Duplikasi (Bab 3.2.2.5) dan operasi mutasi *Aggregate Mate Pool* (Bab 3.2.2.3) telah berhasil memastikan bahwa WGA tidak terjebak dalam *premature convergence* dan *local optimum*.

4.5. Hasil Uji Coba Skenario 2

Sama seperti Uji Coba Skenario 1, Skenario 2 juga menggunakan 7 dataset dan 3 metode petatihan ANN, yaitu ANN-BP, ANN-GA dan ANN-WGA. Semua parameter dan konstanta pada Skenario 2 sama seperti pada Skenario 1. Tabel 4.8 adalah ringkasan hasil Uji Coba Skenario 2.

Tabel 4.8 Ringkasan Hasil Pengujian Skenario 2

Dataset	Jumlah Data			ANN Arch	MdAPE		
	Total	Train	Test		ANN-BP	ANN-GA	ANN-WGA
Liquefaction Dataset 1	1125	844	281	5:9:7:1	5.624	4.404	3.352
Liquefaction Dataset 2	1350	1013	337	5:9:7:1	4.880	4.880	3.487
Liquefaction Dataset 3	2520	1890	630	5:9:7:1	4.377	3.085	2.289
Liquefaction Dataset 4	1200	900	300	5:9:7:1	6.509	6.075	3.730
Liquefaction Dataset 5	1800	1350	450	5:9:7:1	4.161	3.625	2.733
Liquefaction Dataset 6	2100	1575	525	5:9:7:1	3.945	3.100	2.080
Liquefaction Dataset 7	1152	864	288	5:9:7:1	3.327	3.527	2.707
Average MdAPE of All Datasets					4.689	4.100	2.911

Keterangan:

Total = jumlah keseluruhan data

Train = jumlah data pelatihan ANN

Test = jumlah data uji ANN

ANN Arch = konfigurasi jumlah node dalam setiap layer ANN

5:9:7:1 = 5 node input, 9 hidden node + 1 bias node pada hidden layer pertama, 7 hidden node + 1 bias node pada hidden layer kedua, 1 output node

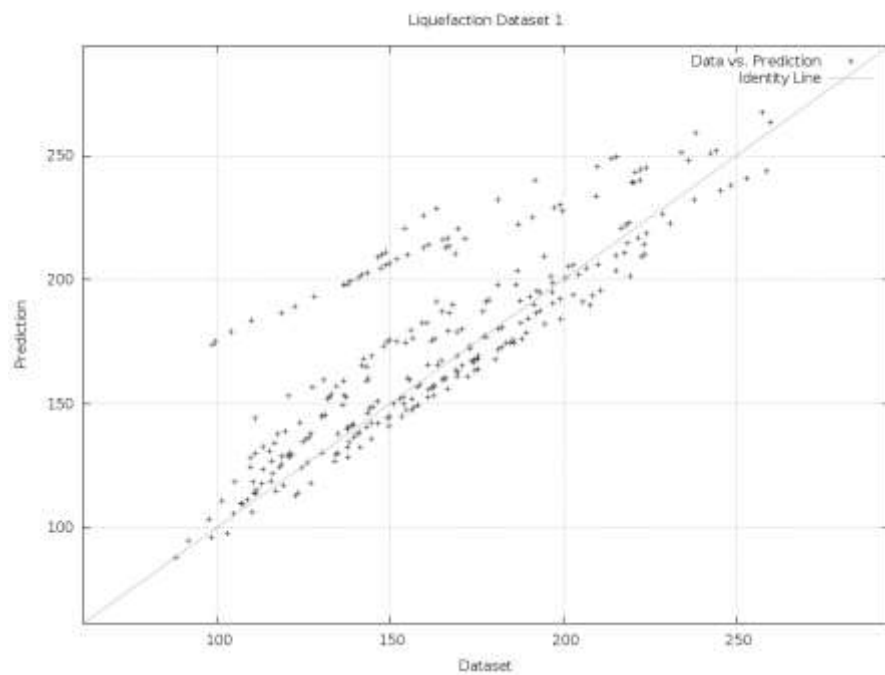
MdAPE = Median Absolute Percentage Error (Bab 2.2.3) dalam persen (%)

Dari hasil uji coba skenarion 2 pada Tabel 4.8 dapat disimpulkan bahwa ANN-WGA menghasilkan MdAPE terkecil 2.080% pada dataset 6 dan MdAPE terbesar 3.730% pada dataset 4. Secara umum untuk semua dataset, kinerja WGA pada uji coba skenario 2 menunjukkan hasil yang lebih baik dibandingkan dengan metode pelatihan BP dan GA. MdAPE ANN-WGA rata-rata sebesar 2.911%.

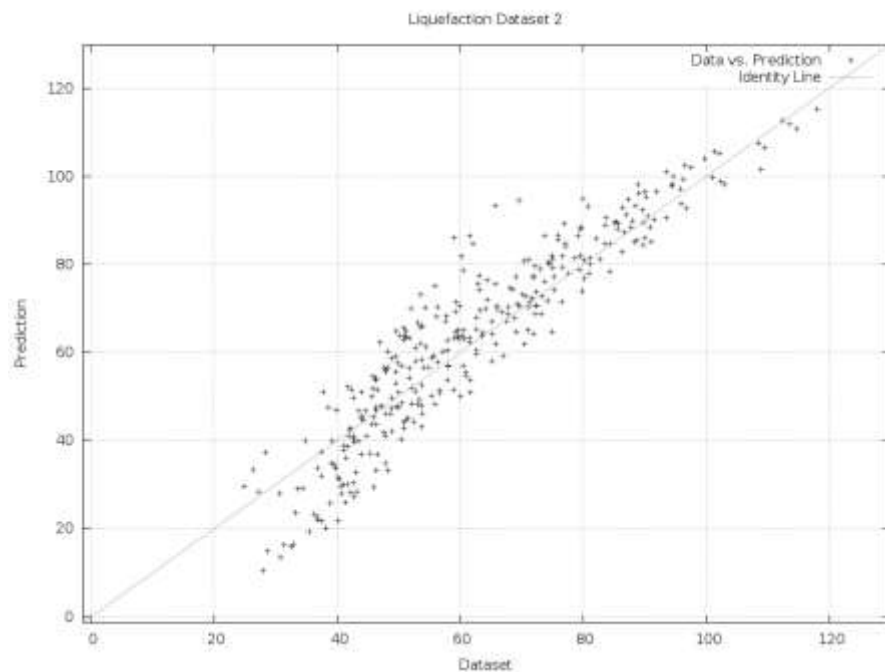
4.5.1. Diagram Scattered Uji Coba Skenario 2

Pada Gambar 4.15 s/d Gambar 4.21 adalah scattered diagram hasil uji coba skenario 2 menggunakan ANN-WGA. Pada Gambar 4.15 penyebaran titik-titik cukup merata sepanjang rentang nilai z , akan tetapi pada terdapat sebagian data yang cukup jauh jaraknya dari garis

identity. Titik-titik yang jauh inilah yang mengakibatkan nilai MdAPE untuk dataset 1 di skenario 2 ini menjadi relatif besar, yaitu 3.352%.

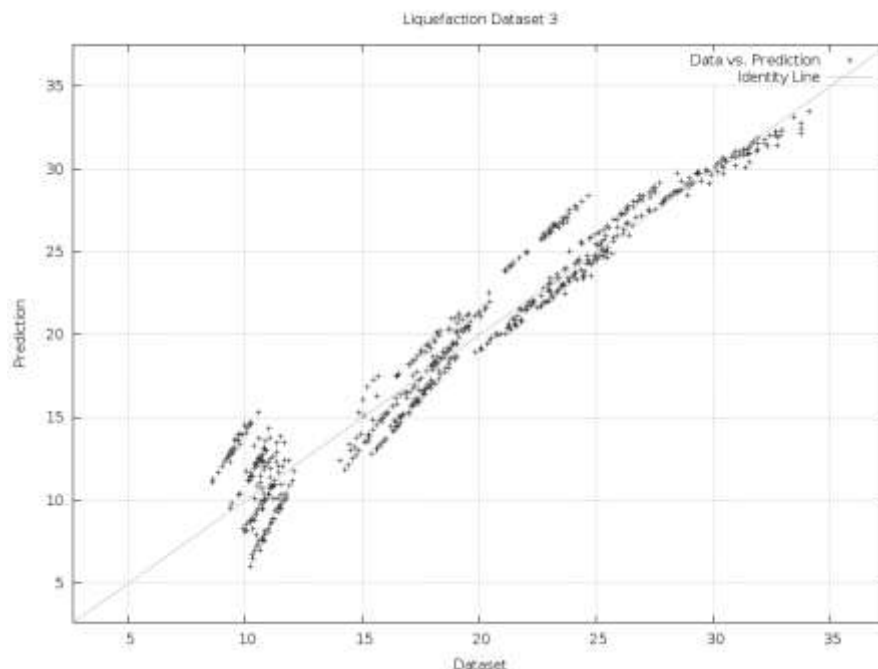


Gambar 4.15 Diagram scattered pada skenario 2 dataset 1



Gambar 4.16 Diagram scattered pada skenario 2 dataset 2

Pada Gambar 4.16, penyebaran titik-titik merata sepanjang rentang nilai z dan jaraknya terhadap garis identity cukup dekat. Hanya sedikit data pada $z < 50m$ yang jaraknya cukup jauh dari garis identity.

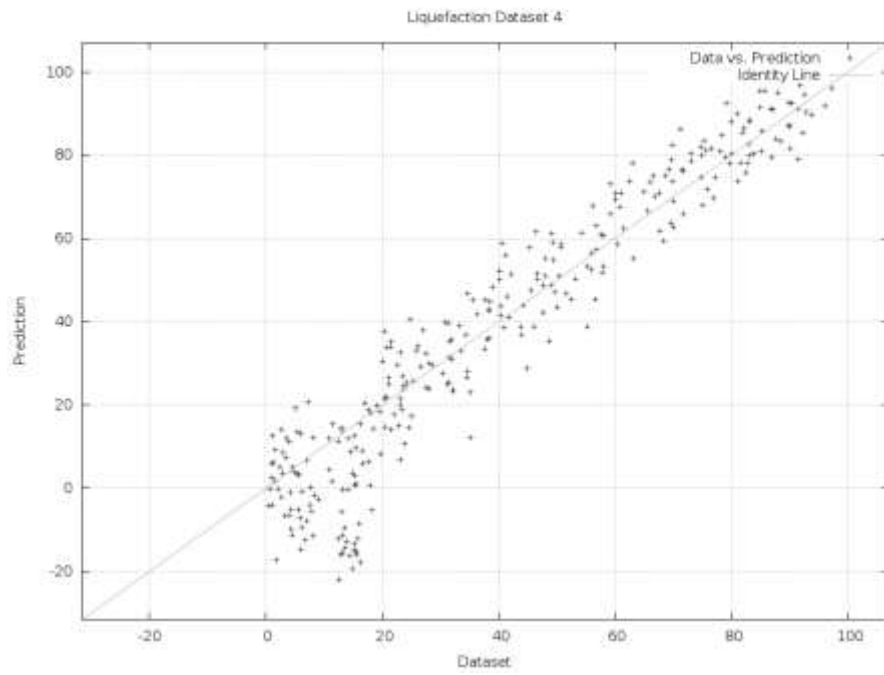


Gambar 4.17 Diagram scattered pada skenario 2 dataset 3

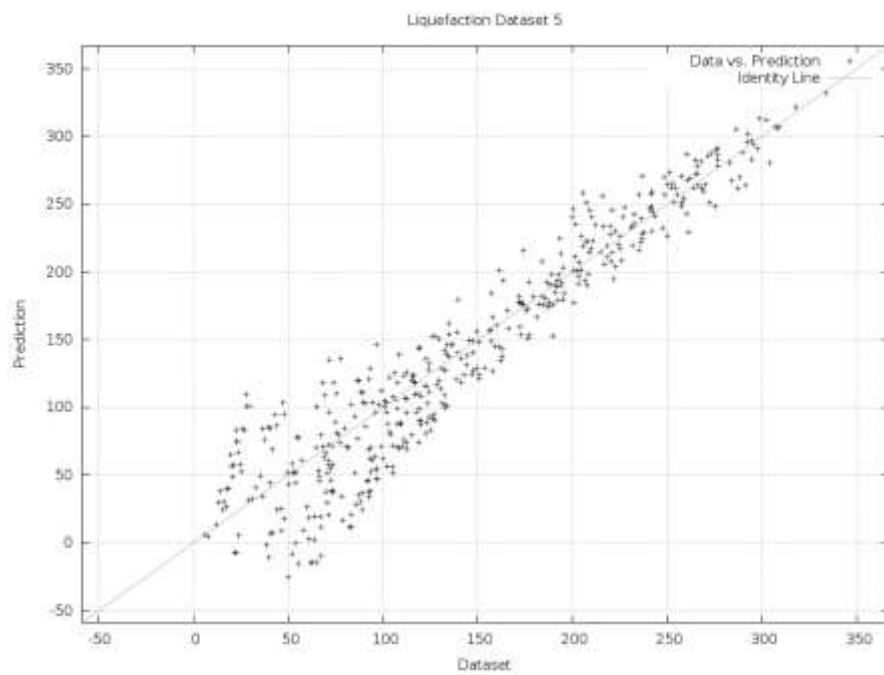
Sama seperti uji coba skenario 1 pada dataset 3 (Gambar 4.3), uji coba skenario 2 pada dataset 3 pada Gambar 4.17 juga mengalami kekosongan nilai z pada rentang $12m < z < 14m$. Penyebaran titik-titik cukup merata sepanjang rentang nilai z dan berjarak cukup dekat dengan garis identity. Uji coba skenario 2 dataset 3 menghasilkan nilai MdAPE yang cukup kecil, yaitu 2.289% (Tabel 4.8).

Pada Gambar 4.18 nampak bahwa penyebaran titik-titik cukup merata sepanjang rentang nilai z dan berjarak cukup dekat dengan garis identity kecuali pada $z < 20m$, beberapa titik berada cukup jauh dari garis identity. Hal ini mengakibatkan nilai MdAPE yang terbesar, yaitu 3.730% bila dibandingkan dengan dataset lain dalam uji coba skenario 2 (Tabel 4.8).

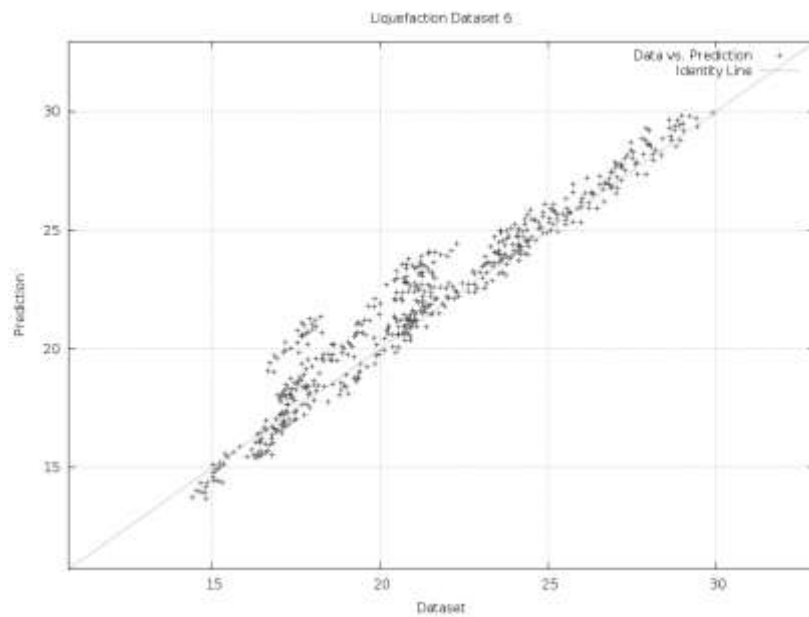
Gambar 4.19 menunjukkan bahwa seluruh penyebaran titik-titik hasil prediksi cukup merata sepanjang rentang nilai z dan berjarak cukup dekat dengan garis identity. Ada beberapa titik yang memiliki jarak cukup jauh dengan garis identity pada rentang $20m < z < 100m$. Hal inilah yang mengakibatkan nilai MdAPE cukup besar, yaitu 2.733% bila dibandingkan dengan dataset 4 dan 6 (Tabel 4.8).



Gambar 4.18 Diagram scattered pada skenario 2 dataset 4

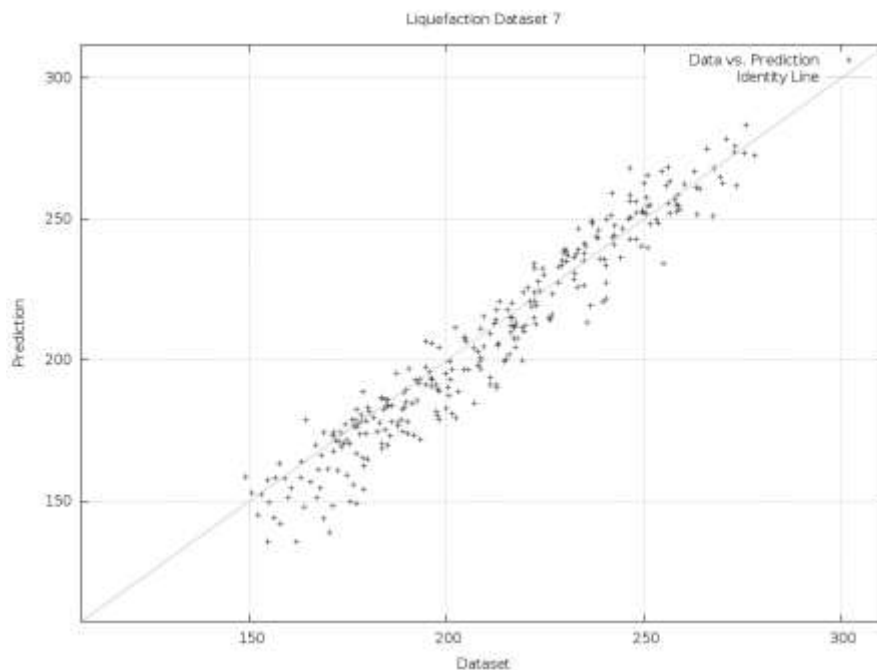


Gambar 4.19 Diagram scattered pada skenario 2 dataset 5



Gambar 4.20 Diagram scattered pada skenario 2 dataset 6

Pada Gambar 4.20, penyebaran titik-titik cukup rapat bahkan ada beberapa titik yang sangat berdekatan. Pola penyebaran juga cukup merata pada semua rentang nilai z . Uji coba skenario 2 dataset 6 menunjukkan hasil MdAPE paling kecil, yaitu 2.080% dengan jumlah data sebesar 2100 (Tabel 4.8). Pola penyebaran yang cukup rapat inilah yang menyebabkan nilai MdAPE menjadi sangat kecil dibandingkan dengan uji coba skenario 2 pada dataset lain.



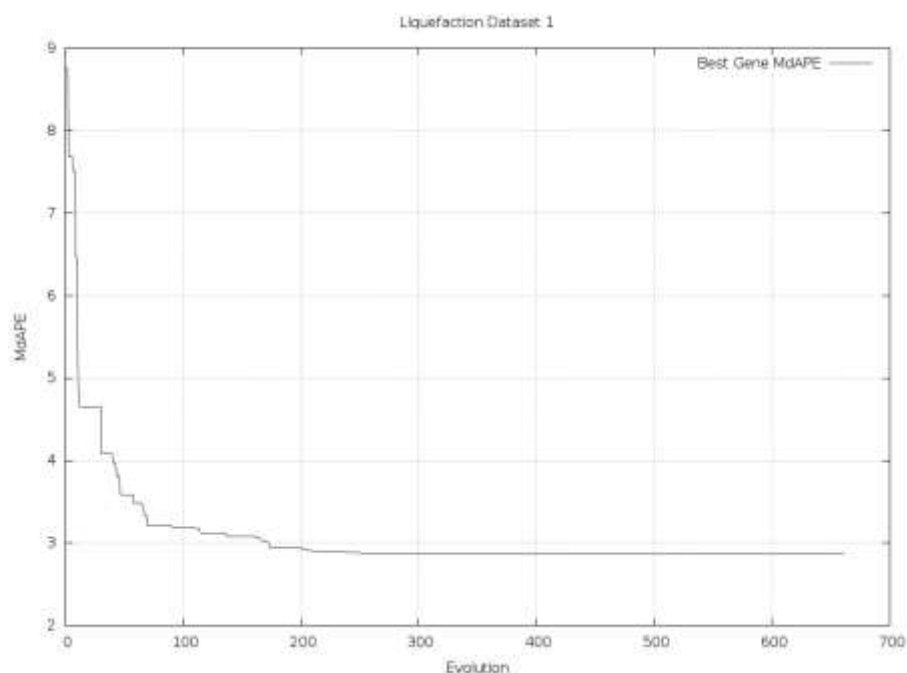
Gambar 4.21 Diagram scattered pada skenario 2 dataset 7

Hasil uji coba skenario 2 dataset 7 pada Gambar 4.21 menunjukkan pola penyebaran yang cukup merata sepanjang rentang nilai z dan berjarak cukup dekat dengan garis identity. MdAPE hasil uji coba sebesar 2.707% terkecil ketiga dalam uji coba skenario 2.

Diagram scattered hasil uji coba skenario 2 pada Gambar 4.15 s/d Gambar 4.21 menunjukkan bahwa hasil setiap prediksi ANN-WGA sudah mendekati garis identity, yang berarti bahwa kesalahan (error) prediksi ANN-WGA sangat kecil. Pola penyebaran titik-titik juga cukup merata di sepanjang rentang nilai z , hal ini berarti bahwa dataset yang digunakan adalah dataset yang cukup baik digunakan sebagai menguji kinerja ANN-WGA terhadap ANN-BP dan ANN-GA.

4.5.2. Diagram Konvergensi Uji Coba Skenario 2

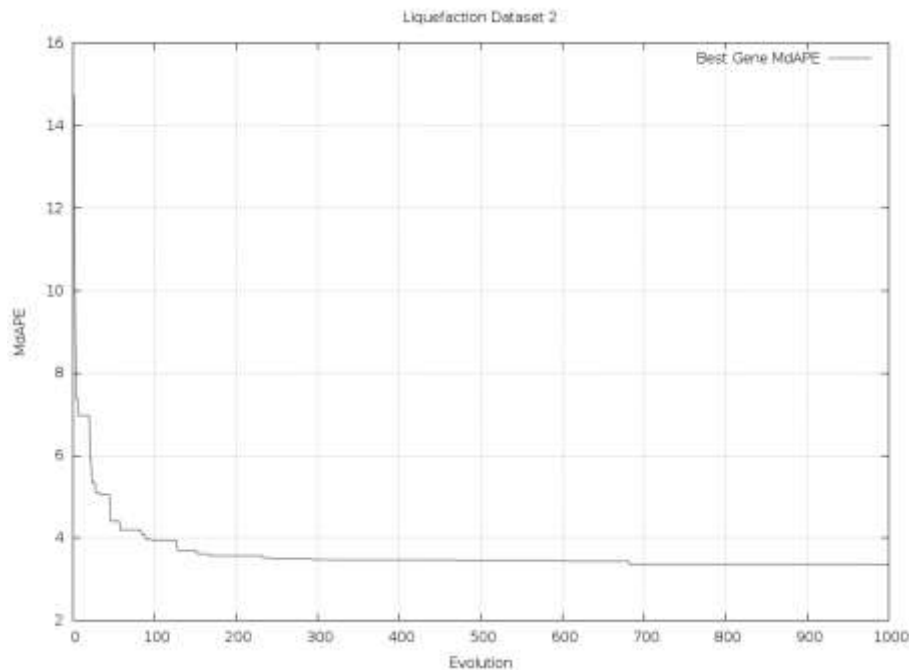
Pada Gambar 4.22 s/d Gambar 4.28 adalah diagram konvergensi untuk uji coba skenario 2 pada dataset 1-7. Diagram konvergensi adalah diagram yang dapat digunakan untuk melihat tingkat konvergensi pada setiap evolusi uji coba yang dilakukan.



Gambar 4.22 Diagram konvergensi pada skenario 2 dataset 1

Pada Gambar 4.22, MdAPE turun secara drastis pada evolusi awal, yaitu pada evolusi < 50 dan kemudian turun secara bertahap sampai evolusi ke-200. Mulai evolusi ke-300 tidak lagi terjadi penurunan MdAPE yang cukup signifikan dan iterasi benar-benar berhenti pada evolusi

ke-670. Iterasi berhenti di evolusi ke-670 karena pemeriksaan konvergensi yang dijelaskan dalam Bab 3.2.2.9. Hal ini berarti bahwa populasi berisi gen-gen yang memiliki nilai *fitness function* MdAPE yang sangat berdekatan walaupun koordinat masing-masing gen cukup jauh jaraknya (Bab 3.2.2.5).

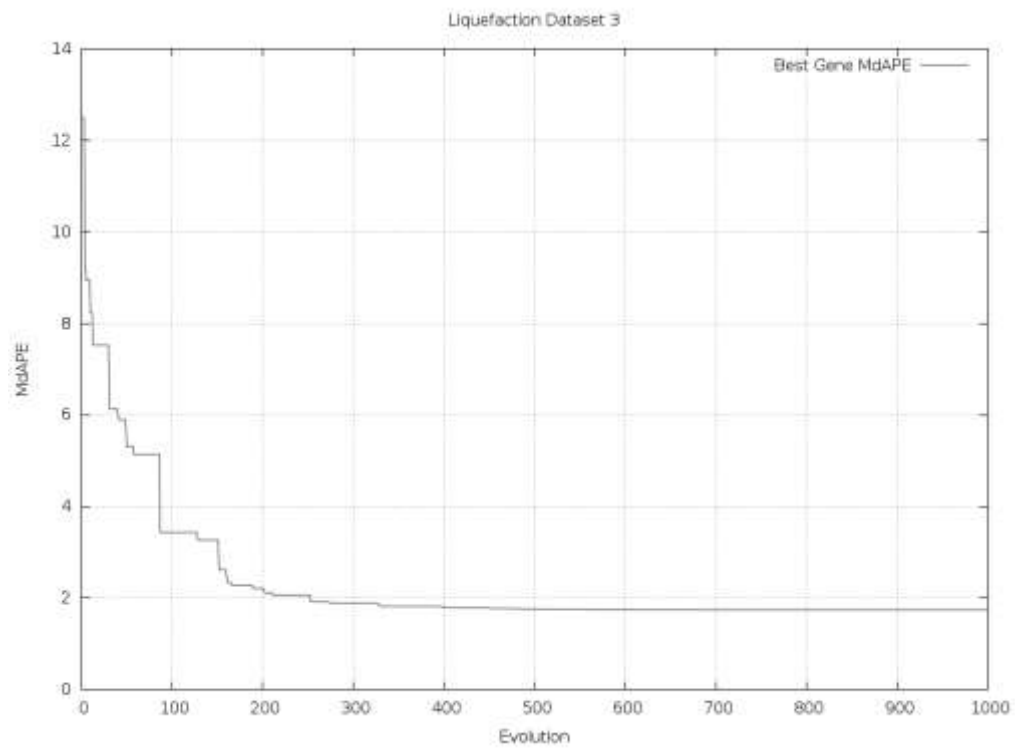


Gambar 4.23 Diagram konvergensi pada skenario 2 dataset 2

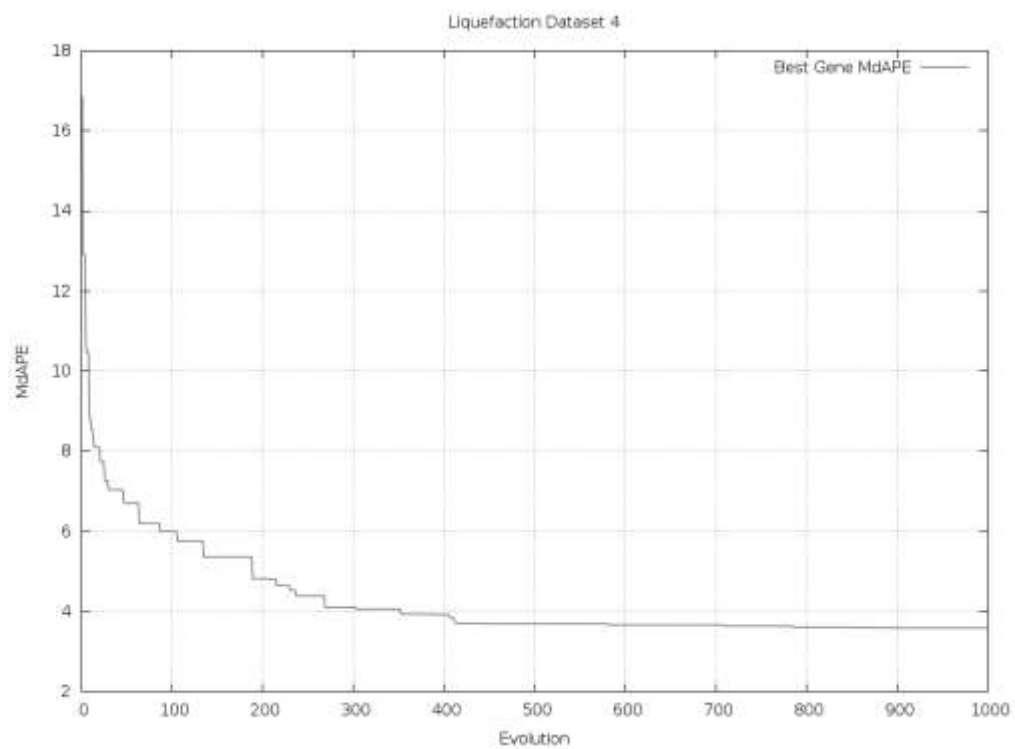
Pada Gambar 4.23, penurunan MdAPE secara drastis terjadi pada awal evolusi, yaitu evolusi ke-20. Selanjutnya penurunan MdAPE terjadi secara perlahan sampai pada evolusi ke-200. Pada evolusi ke-700, tidak lagi terjadi penurunan MdAPE sehingga dapat disimpulkan bahwa konvergensi terjadi pada evolusi ke-700.

Penurunan MdAPE pada Gambar 4.24 terjadi secara perlahan dan lebih *smooth*. Konvergensi mulai terjadi pada evolusi ke-300 dan pada evolusi ke-400 tidak terjadi lagi penurunan MdAPE, sehingga dapat disimpulkan bahwa pada evolusi ke-400 sudah terjadi konvergensi.

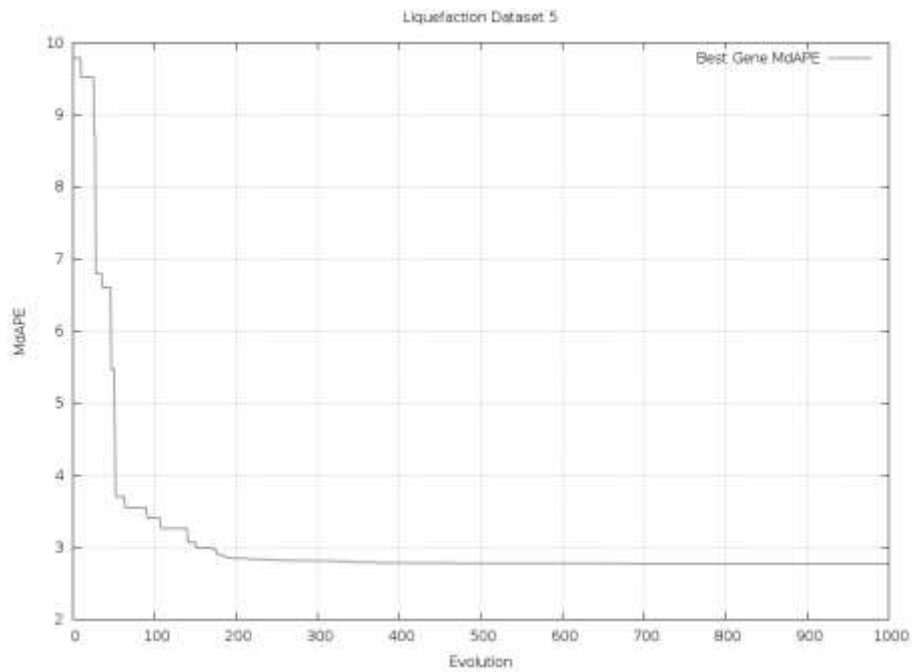
Hasil uji coba skenario 2 dataset 4 pada Gambar 4.25 menunjukkan penurunan MdAPE yang landai dan *smooth*, mirip seperti uji coba skenario 2 dataset 3 (Gambar 4.24). Konvergensi mulai terjadi pada evolusi ke-420 dan pada evolusi ke-800 sudah tidak terjadi lagi penurunan MdAPE. Hal ini dapat disimpulkan bahwa konvergensi telah terjadi pada evolusi ke-800.



Gambar 4.24 Diagram konvergensi pada skenario 2 dataset 3

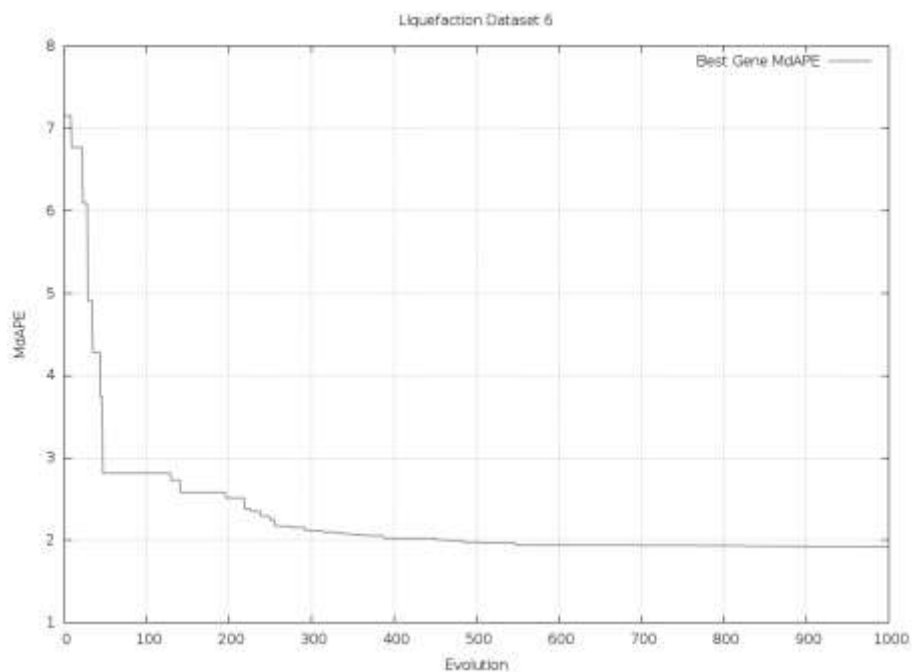


Gambar 4.25 Diagram konvergensi pada skenario 2 dataset 4



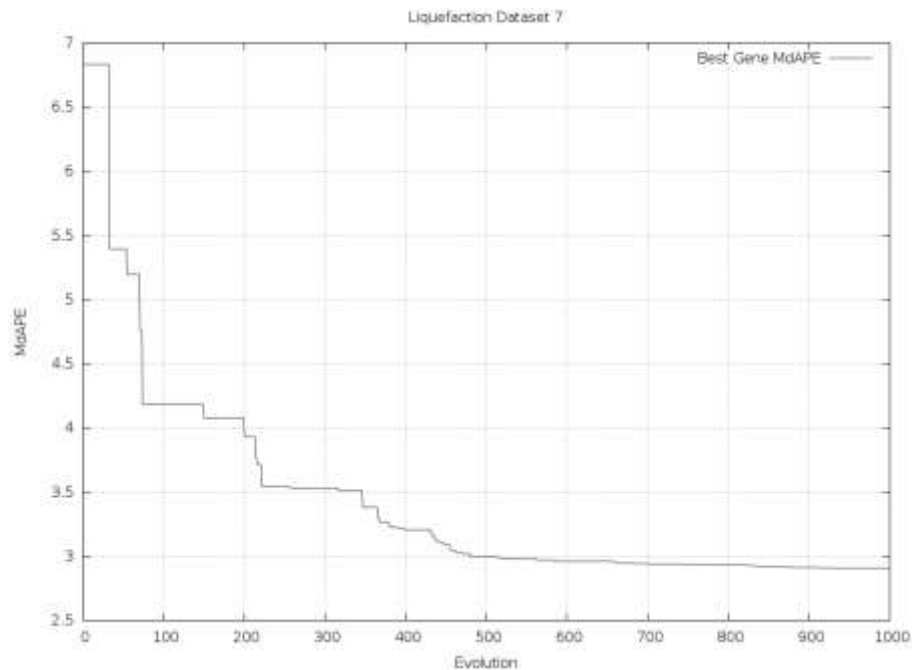
Gambar 4.26 Diagram konvergensi pada skenario 2 dataset 5

Pada Gambar 4.26 lompatan penurunan MdAPE secara drastis terjadi pada awal evolusi, yaitu evolusi < 50 . Kemudian MdAPE turun secara perlahan sampai pada evolusi ke-200. Pada evolusi ke-400, hampir tidak ada lagi penurunan MdAPE sehingga dapat disimpulkan konvergensi terjadi pada evolusi ke-400.



Gambar 4.27 Diagram konvergensi pada skenario 2 dataset 6

Hasil uji coba skenario 2 pada dataset 6 (Gambar 4.27), konvergensi mulai dicapai pada evolusi ke-300 dan pada evolusi ke-600, tidak ada lagi penurunan MdAPE, sehingga dapat disimpulkan bahwa pada evolusi ke-600, sistem ANN-WGA sudah mencapai kondisi konvergen.



Gambar 4.28 Diagram konvergensi pada skenario 2 dataset 7

Pada uji coba skenario 2 pada dataset 7 (Gambar 4.28), konvergensi sudah mulai bisa dicapai sejak evolusi ke-500 dan pada evolusi ke-900 penurunan MdAPE sudah sangat kecil. Hal ini dapat disimpulkan bahwa uji coba skenario 2 pada dataset 7 mencapai kondisi konvergen pada evolusi ke-900.

Diagram konvergensi dari uji coba skenario 2 pada Gambar 4.22 s/d Gambar 4.28 menunjukkan bahwa sebagian besar uji coba mengalami penurunan MdAPE secara drastis pada awal evolusi, kecuali untuk dataset 7 (Gambar 4.28). Seluruh dataset dapat mencapai konvergensi kurang dari jumlah evolusi maksimum, yaitu 1000 iterasi evolusi (Tabel 4.3). Bahkan pada uji coba skenario 2 dataset 1 (Gambar 4.22), iterasi benar-benar berhenti pada evolusi ke-670 karena setiap gen dalam populasi memiliki *fitness function* MdAPE yang mirip sama nilainya (Bab 3.2.2.9). Dari hasil-hasil tersebut dapat disimpulkan bahwa seluruh operasi dalam WGA terutama operasi Anti-Duplikasi (Bab 3.2.2.5), operasi mutasi *Aggregate Mate*

Pool (Bab 3.2.2.3) dan operasi pemeriksaan konvergensi (Bab 3.2.2.9) telah berhasil menghindari *premature convergence* dan *local optimum*.

4.6. Analisa dan Pembahasan Hasil Uji Coba

Setelah melakukan uji coba skenario 1 dan 2 didapatkan beberapa hasil uji coba yang digambarkan dalam diagram scattered dan diagram konvergensi, seperti yang telah diuraikan dalam Bab 4.3 diatas. Tabel 4.9 adalah tabulasi hasil-hasil uji coba skenario 1 dan 2.

Tabel 4.9 Hasil Uji Coba Skenario 1 dan 2

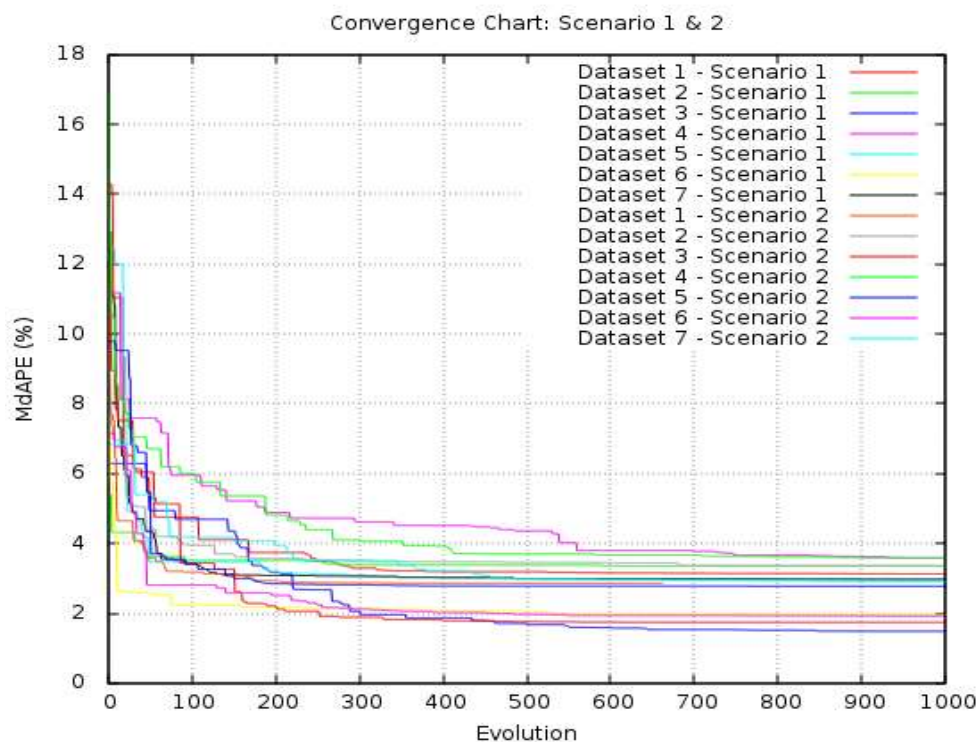
Dataset	Total Data	MdAPE					
		Scenario 1			Scenario 2		
		ANN-BP	ANN-GA	ANN-WGA	ANN-BP	ANN-GA	ANN-WGA
Liquefaction Dataset 1	1,125	5.052	4.065	3.189	5.624	4.404	3.352
Liquefaction Dataset 2	1,350	6.544	4.358	3.241	4.880	4.880	3.487
Liquefaction Dataset 3	2,520	3.032	2.620	1.410	4.377	3.085	2.289
Liquefaction Dataset 4	1,200	6.881	5.082	4.346	6.509	6.075	3.730
Liquefaction Dataset 5	1,800	4.670	3.526	3.050	4.161	3.625	2.733
Liquefaction Dataset 6	2,100	6.257	3.067	2.454	3.945	3.100	2.080
Liquefaction Dataset 7	1,152	5.072	3.760	3.176	3.327	3.527	2.707
Average MdAPE		5.358	3.783	2.981	4.689	4.100	2.911

Berdasarkan Tabel 4.9, berikut ini beberapa analisa hasil uji coba pada skenario 1 dan 2:

1. Uji coba pada dataset 3 dan 6 menunjukkan nilai MdAPE yang terbaik, masing-masing pada urutan pertama dan kedua. Kedua dataset adalah 2 dataset dengan jumlah data paling banyak, yaitu 2520 untuk dataset 3 dan 2100 untuk dataset 1. Untuk dataset lain dengan jumlah dataset yang lebih sedikit selalu menunjukkan hasil MdAPE yang lebih besar dibandingkan MdAPE dataset 3 dan 6. Dari hasil-hasil tersebut dapat disimpulkan bahwa jumlah data dalam sebuah dataset berpengaruh cukup signifikan terhadap kesalahan prediksi (MdAPE).
2. Hasil uji coba pada dataset 4 selalu menunjukkan nilai MdAPE yang terbesar bila dibandingkan dengan dataset lain. Jumlah data dalam dataset 4 relatif kecil bila dibandingkan dengan dataset 3 dan 6. Bila dibandingkan dengan dataset lain yang jumlahnya cukup sama (dataset 1, 2 dan 7), hasil MdAPE uji coba pada dataset 4 masih relatif lebih besar, sehingga bisa disimpulkan bahwa buruknya MdAPE bukan karena jumlah data dalam dataset. Bila diamati kembali spesifikasi dataset pada Bab 4.2 khususnya dataset 4, terdapat nilai kedalaman air (d) yang cukup berbeda dibandingkan

dengan nilai kedalaman air pada dataset lain. Faktor inilah yang menyebabkan dataset 4 menjadi dataset yang paling sulit untuk diprediksi menggunakan ANN.

3. Walaupun dari sudut pandang bidang optimasi, skenario 2 adalah uji coba yang lebih kompleks, akan tetapi rata-rata MdAPE pada uji coba skenario 2 menunjukkan nilai yang tidak jauh berbeda dengan uji coba pada skenario 1. Pada skenario 1 rata-rata MdAPE sebesar 2.981% sementara pada skenario 2 rata-rata MdAPE sebesar 2.911%, perbedaan keduanya hanya terpaut 0.07%. Hal ini berarti bahwa WGA dapat menunjukkan kinerja yang baik dan stabil untuk 2 arsitektur ANN yang berbeda.
4. Hasil dari setiap uji coba yang dilakukan, baik untuk skenario 1 maupun skenario 2, dari dataset 1 s/d dataset 7 menunjukkan bahwa hasil MdAPE untuk ANN-WGA selalu memiliki MdAPE yang terkecil. Hal ini berarti bahwa kinerja WGA jauh lebih baik dan stabil bila dibandingkan dengan BP dan GA konvensional dalam melakukan pelatihan ANN. Dari hasil-hasil tersebut dapat pula disimpulkan bahwa kemampuan WGA telah teruji pada semua dataset, bahkan dataset 4 yang paling sulit sekalipun. Patut diketahui bahwa masing-masing dataset mewakili kondisi alam yang berbeda.



Gambar 4.29 Diagram konvergensi pada skenario 1 dan 2

Diagram konvergensi pada Gambar 4.29 adalah gabungan beberapa diagram konvergensi hasil uji coba skenario 1 dan 2. Berdasarkan Gambar 4.29 tersebut, berikut ini beberapa analisa hasil uji coba pada skenario 1 dan 2 :

1. Pada sebagian besar dataset, kecuali dataset 3 dan 4, MdAPE mengalami penurunan secara drastis, utamanya pada tahap evolusi awal, umumnya terjadi pada evolusi < 100 . Lonjakan penurunan ini disebabkan oleh kombinasi operasi-operasi dalam WGA utamanya operasi *Aggregate Mate Pool Mutation* (AMPM) yang dijelaskan dalam Bab 3.2.2.3. AMPM mampu melakukan lompatan adaptif yang cukup jauh sehingga offspring yang dihasilkan memiliki sifat yang jauh berbeda dengan induknya. Selain AMPM, operasi seleksi Wide Tournament (Bab 3.2.2.1) dan operasi Worst Replacement (Bab 3.2.2.7) mampu memberikan arah baru dalam pencarian solusi.
2. Setelah mengalami penurunan MdAPE yang sangat besar pada tahap awal evolusi, selanjutnya penurunan MdAPE terjadi lebih lambat dan gradual, secara umum terjadi pada rentang evolusi ke-100 s/d evolusi ke-500. Penurunan MdAPE yang semakin kecil diakibatkan oleh semakin kecilnya beda nilai *fitness function* setiap gen dan semakin dekatnya jarak Euclidean antara masing-masing gen. WGA menggunakan operasi anti-duplikasi (Bab 3.2.2.5) sehingga masing-masing gen dalam populasi memiliki jarak Euclidean yang cukup jauh untuk menghindari kondisi *local optimum*. WGA juga menggunakan operasi Worst Replacement (Bab 3.2.2.7), yaitu hanya offspring yang lebih baik dari gen terburuk saja yang bisa masuk dalam populasi utama. Kombinasi operasi anti-duplikasi dan operasi Worst Replacement mengakibatkan jumlah offspring yang masuk ke populasi utama semakin sedikit sehingga memperlambat laju penurunan MdAPE.
3. Tahap akhir konvergensi terjadi manakala penurunan MdAPE terjadi sangat lambat. Pada beberapa dataset tidak lagi mengalami penurunan yang cukup berarti. Hal ini terjadi karena jarak Euclidean antara masing-masing gen sudah sangat dekat, mendekati nilai minimum yang diijinkan (Tabel 4.3 pada Bab 4.3). Secara umum konvergensi terjadi pada rentang evolusi antara evolusi ke-600 s/d evolusi ke-900.

[Halaman ini sengaja dikosongkan]

BAB 5

KESIMPULAN DAN SARAN

Pada Bab 5 ini dijelaskan mengenai kesimpulan akhir yang didapatkan setelah melakukan serangkaian uji coba pada bab sebelumnya.

5.1. Kesimpulan

Berdasarkan dua skenario uji coba yang telah dilakukan dapat diambil beberapa kesimpulan sebagai berikut:

1. ANN-WGA memberikan hasil prediksi yang lebih baik dengan MdAPE yang terkecil dibandingkan ANN-GA dan ANN-BP. Uji coba skenario 1 menghasilkan rata-rata MdAPE sebesar 2.981%, tidak jauh berbeda dengan hasil uji coba skenario 2 dengan rata-rata MdAPE sebesar 2.911%. Kinerja ANN-WGA tetap baik dan stabil untuk arsitektur ANN yang berbeda dan dataset yang berbeda pula.
2. Secara umum, semua dataset dapat konvergen sebelum jumlah iterasi maksimum, yaitu sebesar 1000 evolusi. Tahapan awal evolusi s/d evolusi ke-100, kecepatan konvergensi sangat tinggi ditandai dengan menurunnya nilai MdAPE secara drastis. Setelah evolusi ke-100, konvergensi menjadi lebih lambat. Khusus hanya pada beberapa dataset, konvergensi baru tercapai pada evolusi ke-800 yaitu pada skenario 1 untuk dataset 3 dan 4 serta skenario 2 pada dataset 7.
3. Usulan modifikasi metode optimasi GA yang disebut sebagai WGA mampu menemukan solusi global optimum dalam pelatihan ANN, yaitu mendapatkan konfigurasi bobot koneksi ANN dengan kesalahan prediksi (MdAPE) terkecil.

5.2. Saran

Walaupun WGA telah menunjukkan hasil yang cukup baik dalam uji coba yang telah dilakukan, namun masih terdapat beberapa hal yang bisa dilakukan terkait dengan penelitian ini, antara lain:

1. Dataset yang digunakan dalam penelitian ini diambil dari pembangkitan data sintesis berdasarkan data-data dari penelitian terdahulu. Untuk itu perlu dilakukan studi lebih lanjut menggunakan data pengamatan lingkungan.

2. Konfigurasi ANN yang digunakan dalam penelitian ini berukuran kecil. Untuk itu perlu dilakukan pengujian empiris menggunakan ANN dengan jumlah *hidden layer* dan *hidden node* yang lebih banyak.
3. Dalam bidang ilmu Artificial Intelligence, ada beberapa metode prediksi yang umum digunakan. Kinerja WGA perlu diuji lebih jauh menggunakan teknik hibridasi dengan beberapa metode prediksi lainnya, misalnya SVM (Support Vector Machines).
4. Dalam penelitian ini belum dilakukan studi secara mendalam terutama untuk besaran keragaman populasi. Untuk itu dalam penelitian selanjutnya, besaran keragaman populasi perlu diukur dan digunakan dalam proses evolusi di WGA.

DAFTAR PUSTAKA

- Ariyarit, A. & Kanazaki, M., 2015. Multi-modal distribution crossover method based on two crossing segments bounded by selected parents applied to multi-objective design optimization. *Journal of Mechanical Science and Technology*.
- Bishop, C. M., 2006. *Pattern Recognition and Machine Learning*. New York: Springer Science+Business Media, LLC.
- Cha, D., Blumenstein, M., Zhang, H. & Jeng, D.-S., 2006. *Improvement of an Artificial Neural Network Model using MinMax Preprocessing for the Prediction of Wave-induced Seabed Liquefaction*. Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, IEEE International Joint Conference on Neural Network Proceedings.
- Cha, D. H., Zhang, H., Blumenstein, M. & Jeng, D. S., 2009. Accurate prediction of wave-induced seabed liquefaction at shallow depths using multi-artificial neural networks. *Journal of Coastal Research*, Issue 56.
- Deb, K., 2000. An Efficient Constraint Handling Method for Genetic Algorithms. *Computer methods in applied mechanics and engineering*.
- Diehl, P. U. et al., 2015. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. *International Joint Conference on Neural Networks*, pp. 1-8.
- Gorunescu, F., 2010. *Data Mining: Concepts, Models and Techniques*. Berlin: Springer-Verlag.
- Gupta, D. & Ghafir, S., 2012. An overview of methods maintaining diversity in genetic algorithms. *International journal of emerging technology and advanced engineering*, pp. 56-60.
- Han, J. & Kamber, M., 2006. *Data Mining: Concepts and Techniques*. s.l.:Elsevier.
- Hecht-Nielsen, R., 1987. Kolmogorov's mapping neural network existence theorem. *Proceedings of the international conference on Neural Networks*, Volume 3, pp. 11-13.
- Holland, J. H., 2012. Genetic algorithms. *Scholarpedia*, 2 10, Volume 7, p. 1482.
- Ishihara, K. & Yamazaki, A., 1984. Analysis of wave-induced liquefaction in seabed deposits of sand. *Japanese Society of Soil Mechanics and Foundation Engineering*, 24(3), pp. 85-100.
- Jaky, J., 1944. The coefficient of earth pressure at rest. *Journal of the Society of Hungarian Architects and Engineers*, 78(22), pp. 355-358.
- Jarndal, A., 2015. Combined genetic algorithm and neural network technique for transistor modeling. *ommunications, Signal Processing, and their Applications (ICCSA), 2015 International Conference*.

- Jeng, D. S., 1997. Wave-induced seabed instability in front of a breakwater. *Ocean Engineering*, 24(10), pp. 887-917.
- Larose, D. T., 2005. *Discovering Knowledge in Data: An Introduction to Data Mining*. s.l.:John Wiley & Sons, Inc..
- Lawrence, M. & Patterson, A., 1993. *BrainMaker Professional User's Guide and Reference Manual*. s.l.:California Scientific Software.
- Liao, C. C., Lin, Z., Guo, Y. & Jeng, D.-S., 2015. Coupling model for waves propagating over a porous seabed. *Theoretical and Applied Mechanics Letters*, 5(2), pp. 85-88.
- Liao, C. C., Zhao, H. & Jeng, D.-S., 2015. Poro-Elasto-Plastic Model for the Wave-Induced Liquefaction. *Journal of Offshore Mechanics and Arctic Engineering*, 137(4).
- Lozano, M., Herrera, F. & Ramón, J., 2008. Replacement strategies to preserve useful diversity in steady-state genetic algorithms. *Information Sciences*, 178(23).
- Mirchandani, G. & Wei, C., 1989. On hidden nodes for neural nets. *IEEE Transactions on circuits and systems*, 36(5), pp. 661-664.
- Montana, D. J. & Davis, L., 1989. Training Feedforward Neural Networks Using Genetic Algorithms. *IJCAI*.
- Nair, V. & Hinton, G. E., 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 807-814.
- Nataraja, M. S. & Gill, H. S., 1983. Ocean wave-induced liquefaction analysis. *Journal of Geotechnical Engineering*, 4(109), pp. 573-590.
- Ogasawara, E. et al., 2010. Adaptive normalization: A novel data normalization approach for non-stationary time series. *International Joint Conference on Neural Networks*, pp. 1-8.
- Olson, R., 2014. *Visualization of two dimensions of a NK fitness landscape*. [Online] Available at: <http://bit.ly/2idsjK0> [Accessed 13 October 2016].
- Ortiz, J. M. R., Martínez-Blanco, M. d. R. & René Vega-Carrillo, H., 2011. Evolutionary Artificial Neural Networks in Neutron Spectrometry. *INTECH Open Access Publisher*.
- Park, H. I., 2011. Study for Application of Artificial Neural Networks in Geotechnical Problems. *INTECH Open Access Publisher*.
- Picek, S., Jakobovic, D. & Golub, M., 2013. On the recombination operator in the real-coded genetic algorithms. *Evolutionary Computation (CEC)*.
- Sekiguchi, H., Miyamoto, J., Sassa, S. & Kessel, V., 2004. Progressive solidification of a liquefied sand layer during continued wave loading. *Géotechnique*, 54(10).

- Sekiguchi, H., Sassa, S. & Miyamoto, J., 2001. Analysis of progressive liquefaction as a moving-boundary problem. *Géotechnique*, 51(10).
- Shcherbakov, M. V. et al., 2013. A survey of forecast error measures. *World Applied Sciences Journal*, Issue 24, pp. 171-176.
- Shen, J., 2013. *The Science of Practice: What Happens When You Learn a New Skill*. [Online] Available at: <http://bit.ly/1C2TF9Q> [Accessed October 2016].
- Sumer, B. M., 2014. *Liquefaction Around Marine Structures*. s.l.:s.n.
- Sumer, B. M. & Fredsoe, J., 2005. The Mechanics of Scour in The Marine Environment. In: *Advanced Series Ocean Engineering, Volume 17*. Denmark: World Scientific.
- Takahashi, M. B., Celso, R. J. & Núñez, E. G. F., 2015. Optimization of artificial neural network by genetic algorithm for describing viral production from uniform design data. *Process Biochemistry*.
- Takahashi, M. & Kita, H., 2001. A crossover operator using independent component analysis for real-coded genetic algorithms. *Evolutionary Computation, 2001. Proceedings of the 2001 Congress*.
- Tsutsui, S. & Ghosh, A., 1998. A study on the effect of multi-parent recombination in real coded genetic algorithms. *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference*.
- Tuna, S. C. & Altun, S., 2015. *Modern Approaches in Soil Liquefaction Analysis*. s.l., s.n.
- Yoon, Y. & Kim, Y.-H., 2012. *The roles of crossover and mutation in real-coded genetic algorithms*. s.l.:INTECH Open Access Publisher.
- Zhang, Y., Gao, X. & Katayama, S., 2015. Weld appearance prediction with BP neural network improved by genetic algorithm during disk laser welding. *Journal of Manufacturing Systems*.

[Halaman ini sengaja dikosongkan]

LAMPIRAN 1

HASIL UJI COBA SKENARIO 1: BOBOT ANN

Soil Liquefaction Dataset 1: Best ANN Weight Links

ANN Architecture:

Layer: Input Nodes=5
 Layer: Hidden1 Nodes=9 + 1 Bias
 Layer: Output Nodes=1

Layer Input to Hidden1

--	hidden1-0	hidden1-1	hidden1-2	hidden1-3	hidden1-4	hidden1-5	hidden1-6	hidden1-7	hidden1-8
input-0	-0.13563	0.07437	0.65090	0.29045	0.26158	-0.02088	0.16838	-0.10184	0.43391
input-1	-0.25093	0.49456	-0.07348	-0.14430	0.34503	-0.25160	0.13360	0.33060	-0.18705
input-2	-0.16105	0.11234	-0.00616	-0.13340	0.09767	-1.28191	0.29725	0.38299	0.64685
input-3	-0.03915	0.17651	-0.01858	0.15403	-0.06524	-0.14624	-0.26352	0.21209	0.09558
input-4	0.06110	0.18695	0.49429	0.00927	0.33680	-0.05043	-0.10262	-0.32699	0.33437

Layer Hidden1 to Output

--	output-0
hidden1-0	0.25549
hidden1-1	0.30602
hidden1-2	0.12265
hidden1-3	0.40320
hidden1-4	0.15485
hidden1-5	-0.24829
hidden1-6	0.61386
hidden1-7	0.36191
hidden1-8	-0.00470
bias-1	0.27396

Soil Liquefaction Dataset 2: Best ANN Weight Links

ANN Architecture:

Layer: Input Nodes=5
 Layer: Hidden1 Nodes=9 + 1 Bias
 Layer: Output Nodes=1

Layer Input to Hidden1

--	hidden1-0	hidden1-1	hidden1-2	hidden1-3	hidden1-4	hidden1-5	hidden1-6	hidden1-7	hidden1-8
input-0	0.22207	0.21675	-0.78073	0.47963	-0.08660	0.63656	-0.10758	-0.09388	-0.26839
input-1	0.01075	0.38697	0.77393	0.28615	0.00013	0.11069	-0.39297	-0.13696	0.20706
input-2	0.06660	0.10415	-0.70426	-0.01009	0.75949	0.49511	0.13242	0.03828	0.26062
input-3	-0.16214	-0.07885	-0.44727	0.49059	0.32871	-0.05973	0.15377	0.06740	0.53807
input-4	-0.12522	-0.22477	0.18841	0.03828	0.24505	-0.09588	-0.39832	0.09898	-0.16881

Layer Hidden1 to Output

--	output-0
hidden1-0	0.30222
hidden1-1	0.52214
hidden1-2	0.25631
hidden1-3	0.07008
hidden1-4	0.17662
hidden1-5	0.34296
hidden1-6	0.16651
hidden1-7	0.13716
hidden1-8	0.05837
bias-1	0.21049

Soil Liquefaction Dataset 3: Best ANN Weight Links

ANN Architecture:

Layer: Input Nodes=5
 Layer: Hidden1 Nodes=9 + 1 Bias
 Layer: Output Nodes=1

Layer Input to Hidden1

--	hidden1-0	hidden1-1	hidden1-2	hidden1-3	hidden1-4	hidden1-5	hidden1-6	hidden1-7	hidden1-8
input-0	-1.72019	-0.21344	-0.18177	-0.02969	0.57147	0.26137	-0.52730	1.09126	0.75683
input-1	-0.13453	0.00825	-1.72440	-0.05666	1.02876	-0.32411	0.56851	-0.09949	0.33030
input-2	-0.47126	-0.20167	-0.09567	-0.01390	0.53255	-0.22838	-0.13330	-0.02410	-0.53134
input-3	-0.11661	0.34725	-0.42717	0.61935	0.08242	0.07010	0.04754	0.42557	-0.33786
input-4	0.08639	0.34999	-0.23246	-0.97376	1.10428	0.48289	-0.01886	0.11673	0.25364

Layer Hidden1 to Output

--	output-0
hidden1-0	1.30633
hidden1-1	-0.27381
hidden1-2	-0.00098
hidden1-3	0.10768
hidden1-4	0.11636
hidden1-5	-0.39481
hidden1-6	-0.41968
hidden1-7	0.43641
hidden1-8	0.22299
bias-1	0.27005

Soil Liquefaction Dataset 4: Best ANN Weight Links

ANN Architecture:

Layer: Input Nodes=5
 Layer: Hidden1 Nodes=9 + 1 Bias
 Layer: Output Nodes=1

Layer Input to Hidden1

--	hidden1-0	hidden1-1	hidden1-2	hidden1-3	hidden1-4	hidden1-5	hidden1-6	hidden1-7	hidden1-8
input-0	0.37595	-0.97005	-1.05602	0.34516	0.31045	0.20660	-0.19015	-0.48484	1.11619
input-1	-0.04319	0.28468	-0.86571	-0.27539	-0.46913	-0.05229	-0.54823	0.20127	-0.12531
input-2	-0.31618	-0.41206	-0.65166	-0.37063	-0.17949	1.95441	-0.86583	0.54719	0.02606
input-3	-0.85363	0.67540	0.51255	-0.52070	0.21401	-0.30105	-0.82664	-0.01867	0.39020
input-4	-0.13264	1.60740	0.30919	-0.24828	0.43278	-0.24474	0.41497	0.94702	0.66479

Layer Hidden1 to Output

--	output-0
hidden1-0	-0.23176
hidden1-1	-0.21967
hidden1-2	-0.57869
hidden1-3	-0.32288
hidden1-4	0.09418
hidden1-5	0.05723
hidden1-6	0.28746
hidden1-7	-0.03734
hidden1-8	0.54247
bias-1	-0.27751

Soil Liquefaction Dataset 5: Best ANN Weight Links

ANN Architecture:

Layer: Input Nodes=5
 Layer: Hidden1 Nodes=9 + 1 Bias
 Layer: Output Nodes=1

Layer Input to Hidden1

--	hidden1-0	hidden1-1	hidden1-2	hidden1-3	hidden1-4	hidden1-5	hidden1-6	hidden1-7	hidden1-8
input-0	0.28756	0.10698	-0.55559	1.22693	0.04629	0.31026	0.09495	-0.38909	-0.10890
input-1	-0.19171	0.39692	-0.49817	0.42368	-0.38053	0.24387	0.02809	0.09906	0.09961
input-2	0.86773	-0.26109	0.28188	-0.80057	0.05768	0.33593	-0.82850	-0.24202	0.31577
input-3	0.13210	-0.42044	-0.42294	0.05813	0.32476	-0.23049	-0.02504	-0.43091	-0.28394
input-4	-0.51929	0.46687	1.01356	0.26705	-0.30558	0.92857	-0.14279	-0.51940	-0.29539

Layer Hidden1 to Output

--	output-0
hidden1-0	0.58754
hidden1-1	0.11649
hidden1-2	0.09849
hidden1-3	0.36439
hidden1-4	0.06069
hidden1-5	0.10246
hidden1-6	0.04920
hidden1-7	0.18881
hidden1-8	-0.19003
bias-1	0.63572

Soil Liquefaction Dataset 6: Best ANN Weight Links

ANN Architecture:

Layer: Input Nodes=5
 Layer: Hidden1 Nodes=9 + 1 Bias
 Layer: Output Nodes=1

Layer Input to Hidden1

--	hidden1-0	hidden1-1	hidden1-2	hidden1-3	hidden1-4	hidden1-5	hidden1-6	hidden1-7	hidden1-8
input-0	-0.07786	0.61471	0.37569	0.52643	-0.10812	-0.07759	-0.21092	-0.04324	0.11188
input-1	0.09891	-0.08971	0.50925	-0.01191	0.23114	0.06837	-0.22067	-0.18843	0.03467
input-2	0.13993	0.25445	0.21225	-0.14699	-0.19698	0.15469	-0.17922	-0.14893	-0.22625
input-3	-0.19318	-0.29831	-0.04591	-0.10187	-0.00702	-0.50777	0.34010	-0.28639	1.31686
input-4	-0.70561	-0.53789	-0.00588	-0.09138	-1.27591	0.02518	0.02865	-0.16827	0.00416

Layer Hidden1 to Output

--	output-0
hidden1-0	0.37053
hidden1-1	0.00988
hidden1-2	0.80953
hidden1-3	0.38305
hidden1-4	0.03770
hidden1-5	0.00584
hidden1-6	-0.02201
hidden1-7	0.14356
hidden1-8	0.05393
bias-1	-0.17148

Soil Liquefaction Dataset 7: Best ANN Weight Links

ANN Architecture:

Layer: Input Nodes=5

Layer: Hidden1 Nodes=9 + 1 Bias

Layer: Output Nodes=1

Layer Input to Hidden1

--	hidden1-0	hidden1-1	hidden1-2	hidden1-3	hidden1-4	hidden1-5	hidden1-6	hidden1-7	hidden1-8
input-0	0.23687	-1.02063	0.46194	0.62753	-1.59846	0.04490	0.09294	-0.33199	-0.12071
input-1	0.13606	0.50670	0.20017	-0.49644	0.42067	1.30251	0.28912	-0.50674	-1.24978
input-2	-0.34027	0.25841	0.24091	-0.31907	-0.23591	-1.08878	0.31914	-0.28818	-0.46876
input-3	1.63334	0.41339	-0.11274	0.00237	0.32452	-0.06282	1.11343	0.19248	0.11202
input-4	-0.31280	0.63813	0.11295	-0.21443	2.69106	-0.41023	0.40031	0.54811	-0.40453

Layer Hidden1 to Output

--	output-0
hidden1-0	-0.14825
hidden1-1	-0.25248
hidden1-2	0.64490
hidden1-3	1.07691
hidden1-4	-0.08264
hidden1-5	0.04338
hidden1-6	0.41545
hidden1-7	0.74926
hidden1-8	-0.01241
bias-1	-0.52998

LAMPIRAN 2

HASIL UJI COBA SKENARIO 2: BOBOT ANN

Soil Liquefaction Dataset 1: Best ANN Weight Links

ANN Architecture:

Layer: Input Nodes=5
 Layer: Hidden1 Nodes=9 + 1 Bias
 Layer: Hidden2 Nodes=7 + 1 Bias
 Layer: Output Nodes=1

Layer Input to Hidden1

--	hidden1-0	hidden1-1	hidden1-2	hidden1-3	hidden1-4	hidden1-5	hidden1-6	hidden1-7	hidden1-8
input-0	0.20748	-0.73754	0.02939	0.41638	0.06446	-0.29462	-0.89993	-0.29313	0.31672
input-1	-0.07206	-0.04900	-0.07649	0.72611	-0.01120	0.46544	-0.29619	-0.81607	0.80386
input-2	-0.16510	0.37911	-0.02539	0.30867	-0.64430	-0.61093	-0.14316	-0.44517	0.45135
input-3	-0.40647	0.02487	0.46652	-0.14706	-0.12114	-0.33788	-0.24344	0.19285	-0.19672
input-4	-0.71514	-0.41317	-0.00374	-0.30464	-1.88130	-0.23032	0.12783	0.20483	0.12703

Layer Hidden1 to Hidden2

--	hidden2-0	hidden2-1	hidden2-2	hidden2-3	hidden2-4	hidden2-5	hidden2-6
hidden1-0	0.67982	-0.68002	0.33687	0.06824	0.73807	0.25182	0.02512
hidden1-1	0.31860	0.33850	0.43815	0.64776	0.87885	-0.07970	-0.13278
hidden1-2	0.91072	-1.18285	-0.24537	0.85061	-0.40761	-0.07552	-0.10745
hidden1-3	0.21866	0.24481	-0.16382	-0.03881	-0.26908	0.98864	-0.55106
hidden1-4	0.42454	0.17601	-0.16534	0.95802	0.54392	0.23835	0.33726
hidden1-5	0.16416	-0.00834	-0.57273	0.11923	-0.22138	-0.20751	-0.16751
hidden1-6	-0.64768	0.74028	-0.24165	0.08488	-1.30643	-0.43211	0.67816
hidden1-7	0.24472	-0.27183	-0.11751	-0.01954	0.19360	-0.27312	-0.36995
hidden1-8	1.29626	-0.26764	0.06215	-0.93466	-0.11420	0.10411	-0.46484
bias-1	-0.55072	0.37103	0.39594	0.20828	1.22916	0.27419	0.06198

Layer Hidden2 to Output

--	output-0
hidden2-0	0.29128
hidden2-1	-0.18600
hidden2-2	-0.41138
hidden2-3	0.16567
hidden2-4	-0.29367
hidden2-5	0.25404
hidden2-6	0.07666
bias-2	-0.38012

Soil Liquefaction Dataset 2: Best ANN Weight Links

ANN Architecture:

Layer: Input Nodes=5
 Layer: Hidden1 Nodes=9 + 1 Bias
 Layer: Hidden2 Nodes=7 + 1 Bias
 Layer: Output Nodes=1

Layer Input to Hidden1

--	hidden1-0	hidden1-1	hidden1-2	hidden1-3	hidden1-4	hidden1-5	hidden1-6	hidden1-7	hidden1-8
input-0	-0.22156	-0.13654	0.92550	0.05568	0.30219	0.11166	0.08447	0.72541	0.82909
input-1	0.07674	-0.09423	0.26696	-0.12488	-0.12383	-0.48247	0.31779	0.36621	-0.58373
input-2	-0.18052	-1.94357	0.00762	-0.31031	0.48489	0.32998	0.07823	0.82725	-0.16653
input-3	-0.45555	0.17791	0.44399	0.37016	0.21882	-0.15657	-0.28570	0.17966	-0.54949
input-4	-0.28625	-0.37704	0.32096	0.72598	-0.00126	-0.32009	-0.17637	0.05857	-0.02561

Layer Hidden1 to Hidden2

--	hidden2-0	hidden2-1	hidden2-2	hidden2-3	hidden2-4	hidden2-5	hidden2-6
hidden1-0	0.73607	-0.07701	-0.25049	-0.14485	-0.37043	-0.04736	0.68807
hidden1-1	0.90221	0.45676	-0.21965	0.28655	0.72328	-0.08454	-0.28670
hidden1-2	-0.31383	0.34969	-0.14890	-1.35366	0.05123	0.11789	-0.10860
hidden1-3	-0.47920	-0.77872	0.59228	0.11710	-0.08715	0.03755	0.16825
hidden1-4	-0.67199	0.07668	0.06082	-0.18143	-0.30145	-0.18177	-0.18082
hidden1-5	0.31870	-0.61162	-0.02644	-0.20215	-0.04236	-0.15635	0.39139
hidden1-6	-0.48793	0.04680	-0.08196	-0.11972	0.09601	0.22006	0.86353
hidden1-7	0.21942	1.27157	-0.36365	-0.18154	-0.08258	-0.02805	-0.23816
hidden1-8	-0.39891	-0.28103	0.40015	0.79941	-0.09924	-0.43255	-0.09245
bias-1	-0.90389	0.38487	-1.38468	-0.12586	-0.23929	0.33968	0.23856

Layer Hidden2 to Output

--	output-0
hidden2-0	-0.12725
hidden2-1	0.31751
hidden2-2	-0.20272
hidden2-3	0.05503
hidden2-4	0.53368
hidden2-5	0.25293
hidden2-6	0.50622
bias-2	-0.22625

Soil Liquefaction Dataset 3: Best ANN Weight Links

ANN Architecture:

Layer: Input Nodes=5
 Layer: Hidden1 Nodes=9 + 1 Bias
 Layer: Hidden2 Nodes=7 + 1 Bias
 Layer: Output Nodes=1

Layer Input to Hidden1

--	hidden1-0	hidden1-1	hidden1-2	hidden1-3	hidden1-4	hidden1-5	hidden1-6	hidden1-7	hidden1-8
input-0	0.17351	-0.40850	0.61422	0.78241	0.33264	0.30121	0.48081	-0.55019	0.16095
input-1	0.02617	-0.75055	-0.07228	0.19787	-0.82121	0.90790	0.42552	0.49504	-0.52842
input-2	0.28468	0.58027	0.27591	-0.03006	-0.08915	-0.21476	-0.18208	1.19260	-0.60047
input-3	0.46802	-0.08422	0.19008	-0.16181	0.07248	-0.92140	-0.47297	-2.02191	0.03275
input-4	0.08345	-0.12187	-0.44751	0.25328	-0.06537	-0.40440	-0.04482	-0.82840	-0.39606

Layer Hidden1 to Hidden2

--	hidden2-0	hidden2-1	hidden2-2	hidden2-3	hidden2-4	hidden2-5	hidden2-6
hidden1-0	0.31879	0.77755	0.28350	0.16269	-0.79661	-0.37618	-0.61959
hidden1-1	-0.45870	0.17492	-0.43462	-0.44258	0.05524	0.59609	0.06451
hidden1-2	0.24080	0.66346	-0.40235	0.78958	-0.04700	-0.59898	-0.40969
hidden1-3	0.43210	-0.17160	-1.03387	-0.08912	-0.05784	-0.02764	-0.24422
hidden1-4	-0.36044	-0.92565	-0.15213	-0.33243	0.19562	0.33491	-1.77919
hidden1-5	0.04819	-0.01915	0.66845	-0.71067	1.36340	-0.10133	-0.70632
hidden1-6	0.47367	0.54103	0.29956	-0.05297	0.07424	-0.96688	0.37312
hidden1-7	0.12018	-0.37280	0.39415	0.03050	1.21566	-0.06436	0.58602
hidden1-8	0.24325	-0.33643	-0.05936	-0.71448	-0.08599	-0.20231	0.81044
bias-1	-0.00666	-0.03372	-0.66885	0.84801	0.52506	0.42161	-1.24121

Layer Hidden2 to Output

--	output-0
hidden2-0	0.78929
hidden2-1	0.08404
hidden2-2	0.07617
hidden2-3	0.13136
hidden2-4	0.67295
hidden2-5	-0.24055
hidden2-6	-0.30158
bias-2	0.29129

Soil Liquefaction Dataset 4: Best ANN Weight Links

ANN Architecture:

Layer: Input Nodes=5
 Layer: Hidden1 Nodes=9 + 1 Bias
 Layer: Hidden2 Nodes=7 + 1 Bias
 Layer: Output Nodes=1

Layer Input to Hidden1

--	hidden1-0	hidden1-1	hidden1-2	hidden1-3	hidden1-4	hidden1-5	hidden1-6	hidden1-7	hidden1-8
input-0	0.04240	0.38439	0.96554	1.03113	0.03667	0.08395	0.04715	0.28137	-0.01265
input-1	0.08457	0.24197	-0.37241	0.01603	0.03455	0.38473	0.02543	0.08964	-0.07193
input-2	-0.09951	0.05830	-0.00845	0.45315	0.23324	0.06193	-0.24636	0.12837	0.02088
input-3	-0.67359	0.03567	0.26381	0.06826	-0.13854	-0.04560	-0.56504	-0.48289	0.22322
input-4	-0.02671	-0.00684	0.09772	-0.15316	0.29296	0.10969	0.72394	-0.06456	0.16598

Layer Hidden1 to Hidden2

--	hidden2-0	hidden2-1	hidden2-2	hidden2-3	hidden2-4	hidden2-5	hidden2-6
hidden1-0	0.69187	0.11720	-0.12496	0.07070	0.46038	-0.15495	-0.00751
hidden1-1	0.02596	-0.05920	0.32544	-0.23577	0.04358	0.15407	0.34227
hidden1-2	-0.06732	0.42281	0.06316	0.14365	-0.16753	-0.02753	0.64728
hidden1-3	0.19749	0.23747	0.27305	-0.27636	0.06150	0.13605	0.38797
hidden1-4	0.00615	0.05929	0.00704	0.36085	0.36273	-0.07168	0.07295
hidden1-5	-0.00724	-0.54005	0.01546	0.24121	-0.77821	-0.10276	0.06646
hidden1-6	0.05054	-0.20323	0.34320	-0.34233	-0.67348	0.37697	0.19930
hidden1-7	0.15495	-0.24605	0.63586	-0.14519	0.13421	0.12460	0.16931
hidden1-8	0.31981	-0.21068	-0.15016	0.00779	-0.02936	-0.08055	-0.01784
bias-1	0.09189	0.05924	0.21061	-0.38804	-0.08241	0.03759	0.01166

Layer Hidden2 to Output

--	output-0
hidden2-0	-0.02810
hidden2-1	0.47135
hidden2-2	0.31972
hidden2-3	0.04158
hidden2-4	0.44006
hidden2-5	0.00651
hidden2-6	0.38550
bias-2	0.47877

Soil Liquefaction Dataset 5: Best ANN Weight Links

ANN Architecture:

Layer: Input Nodes=5
 Layer: Hidden1 Nodes=9 + 1 Bias
 Layer: Hidden2 Nodes=7 + 1 Bias
 Layer: Output Nodes=1

Layer Input to Hidden1

--	hidden1-0	hidden1-1	hidden1-2	hidden1-3	hidden1-4	hidden1-5	hidden1-6	hidden1-7	hidden1-8
input-0	-0.80996	-0.89671	0.41846	0.26441	1.11649	1.09934	-0.13919	0.48044	-0.23854
input-1	-0.20944	0.54394	0.72026	-0.78557	-0.67527	0.30821	-1.17719	-0.08966	-1.29195
input-2	0.55622	-0.22566	-0.31810	-0.43620	-0.23001	0.20971	-0.30506	0.24835	-0.51723
input-3	0.36825	-0.79796	-0.27836	0.01484	-0.15764	-0.03427	0.14191	0.03544	0.26870
input-4	0.02278	-0.06454	-0.10306	-0.10931	-0.22230	0.01160	0.10866	-0.36559	-0.10271

Layer Hidden1 to Hidden2

--	hidden2-0	hidden2-1	hidden2-2	hidden2-3	hidden2-4	hidden2-5	hidden2-6
hidden1-0	-0.59455	-0.49170	-0.26685	-0.09574	0.11997	0.11550	-1.29718
hidden1-1	-0.32676	-0.82700	-0.66992	-0.00247	-0.26669	-0.09442	2.05464
hidden1-2	-0.41657	-0.11035	0.53620	0.49519	0.30746	-0.54832	-0.44178
hidden1-3	-0.29719	-0.41930	-0.75710	0.30962	-0.99273	-0.15548	0.98082
hidden1-4	0.13326	-0.88993	-2.22114	-0.18149	0.01181	-0.26380	-0.13210
hidden1-5	-0.29741	-0.17169	-0.54169	0.45699	0.29076	0.56396	-0.89746
hidden1-6	0.23672	0.39823	0.61084	0.29931	0.55457	-0.18833	-0.50110
hidden1-7	0.05561	-0.05413	-0.61391	-0.36804	0.42981	0.16301	-1.58032
hidden1-8	0.74830	-0.01378	-0.04161	0.41932	-0.33412	-0.12518	0.11208
bias-1	0.04658	-0.00022	0.50441	-0.24286	0.70488	0.84732	0.59211

Layer Hidden2 to Output

--	output-0
hidden2-0	0.08789
hidden2-1	-0.44920
hidden2-2	-0.48474
hidden2-3	0.17568
hidden2-4	0.55787
hidden2-5	0.57370
hidden2-6	-0.64996
bias-2	0.48483

Soil Liquefaction Dataset 6: Best ANN Weight Links

ANN Architecture:

Layer: Input Nodes=5
 Layer: Hidden1 Nodes=9 + 1 Bias
 Layer: Hidden2 Nodes=7 + 1 Bias
 Layer: Output Nodes=1

Layer Input to Hidden1

--	hidden1-0	hidden1-1	hidden1-2	hidden1-3	hidden1-4	hidden1-5	hidden1-6	hidden1-7	hidden1-8
input-0	0.21456	0.33916	0.00234	0.12767	0.75216	0.29397	-0.02329	0.27207	-0.21594
input-1	0.20146	-0.20296	0.17635	-0.38117	0.72122	-0.07556	0.24080	-0.05350	-0.28525
input-2	0.16022	-0.27059	0.69632	-0.34835	-0.21285	0.04074	-0.16244	0.08976	0.17155
input-3	0.22775	-0.19670	0.29748	0.03077	-0.03001	-0.26996	-0.04411	-0.16453	-0.09447
input-4	0.01340	-0.48568	-0.18838	0.06644	-0.32633	0.49423	-0.54681	0.00654	0.13287

Layer Hidden1 to Hidden2

--	hidden2-0	hidden2-1	hidden2-2	hidden2-3	hidden2-4	hidden2-5	hidden2-6
hidden1-0	-0.41458	-0.64496	0.22394	0.13222	-0.60017	0.37497	0.47584
hidden1-1	0.06325	0.36745	-0.21549	-0.80749	-0.00934	0.35673	0.13538
hidden1-2	-0.19988	0.30998	0.29184	-0.40590	-0.07868	-0.04063	-0.08200
hidden1-3	0.45551	-0.56023	-0.03255	0.01135	0.02604	-0.03375	-0.58328
hidden1-4	-0.08084	0.54536	0.52726	0.50453	0.30818	0.16350	0.51337
hidden1-5	-0.92816	0.16211	0.31938	0.00608	0.06261	-0.02805	0.12641
hidden1-6	-0.25267	0.57588	0.56402	0.20050	0.17481	0.50500	0.20420
hidden1-7	-0.54694	0.69923	0.46439	-0.15416	-0.34041	0.09584	0.27378
hidden1-8	-0.46786	0.59015	-0.67920	0.29743	0.34738	-0.20019	2.30420
bias-1	0.26281	-0.08522	0.86381	0.15034	0.62512	0.55559	-0.09778

Layer Hidden2 to Output

--	output-0
hidden2-0	-0.44976
hidden2-1	-0.11455
hidden2-2	0.73590
hidden2-3	0.01241
hidden2-4	0.78274
hidden2-5	0.17483
hidden2-6	0.11224
bias-2	0.26656

Soil Liquefaction Dataset 7: Best ANN Weight Links

ANN Architecture:

Layer: Input Nodes=5
 Layer: Hidden1 Nodes=9 + 1 Bias
 Layer: Hidden2 Nodes=7 + 1 Bias
 Layer: Output Nodes=1

Layer Input to Hidden1

--	hidden1-0	hidden1-1	hidden1-2	hidden1-3	hidden1-4	hidden1-5	hidden1-6	hidden1-7	hidden1-8
input-0	0.19739	-0.80686	0.04198	0.00272	-0.36843	0.64622	0.48206	0.24715	0.10136
input-1	0.30259	-0.24314	0.06724	-1.26271	-0.06313	-0.42857	0.52185	0.18523	0.06296
input-2	0.16965	0.19388	-0.66290	0.52732	-0.23968	0.47666	-0.04594	-0.82537	-0.69770
input-3	-0.60174	-0.16157	-0.36791	0.10166	0.52374	0.25266	0.31751	0.71515	0.26912
input-4	0.62124	-0.57954	-0.58678	-0.34174	-0.04448	-0.49625	-0.21678	-1.08581	0.32622

Layer Hidden1 to Hidden2

--	hidden2-0	hidden2-1	hidden2-2	hidden2-3	hidden2-4	hidden2-5	hidden2-6
hidden1-0	0.11426	-0.04478	0.14231	0.48276	0.24035	0.42425	-0.52584
hidden1-1	0.09493	-0.14835	-0.01839	0.24195	-0.32485	-0.49329	-0.10575
hidden1-2	-0.14020	-0.67643	-0.65390	0.63951	0.38624	-0.26012	2.96619
hidden1-3	-0.67810	-0.40414	0.44690	0.12972	-1.39172	0.19732	0.62168
hidden1-4	-0.18045	-0.33741	-0.10248	0.47777	0.09648	0.67864	0.68317
hidden1-5	-0.69859	0.68215	0.61675	0.32646	-0.29447	0.65091	0.16197
hidden1-6	0.38926	-0.53151	0.10420	0.68243	0.08719	0.50836	0.03599
hidden1-7	-0.35175	0.33051	-0.24455	-0.45704	-0.13436	-0.03982	0.37577
hidden1-8	-0.37108	0.28816	-0.12388	0.16063	-0.22844	0.29874	-0.21098
bias-1	0.60750	0.80237	-0.34661	0.08779	-0.30643	-0.66216	0.70227

Layer Hidden2 to Output

--	output-0
hidden2-0	-0.24755
hidden2-1	0.08257
hidden2-2	0.38270
hidden2-3	0.28231
hidden2-4	0.32692
hidden2-5	0.38597
hidden2-6	0.01390
bias-2	-0.63460